AFRL-RH-WP-TR-2016-0074

ACTIVE LEARNING FOR AUTOMATIC AUDIO PROCESSING OF UNWRITTEN LANGUAGES (ALAPUL)

Dimitra Vergyri
Andreas Kathol
Wen Wang
Chris Bartels
Julian VanHout
Vikramjit Mitra
Colleen Richey

SRI International
Information and Computing Sciences Division
Speech Technology and Research Laboratory
333 Ravenswood Ave.
Menlo Park, CA 94025-3493

JULY 2016

Final Report

AIR FORCE RESEARCH LABORATORY
711TH HUMAN PERFORMANCE WING,
AIRMAN SYSTEMS DIRECTORATE,
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th Air Base Wing Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RH-WP-TP-2016-0074 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//Signed//	//Signed//
WILLIAM KILPATRICK, Ph.D., WUM	LOUISE A. CARTER, Ph.D., DR-IV
Human Trust and Interaction Branch	Chief, Human Centered ISR Division
Airman Systems Directorate	Airman Systems Directorate
711th Human Performance Wing	711th Human Performance Wing
Air Force Research Laboratory	Air Force Research Laboratory

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
22-07-2016	Final	June 2015-July 2016

4. TITLE AND SUBTITLE
Active Learning for Automatic Audio Processing of Unwritten Languages (ALAPUL)

5a. CONTRACT NUMBER
FA8650-15-C-9101

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
*Dimitra Vergyri; Andreas Kathol; Wen Wang; Chris Bartels; Julian VanHout; Vikramjit Mitra; Colleen Richey.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
H0L4

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
*SRI International
Information and Computing Sciences Division
Speech Technology and Research Laboratory
333 Ravenswood Ave.
Menlo Park, CA 94025-3493

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Material Command
Air Force Research Laboratory
Human Performance Wing
Airman Systems Directorate
Human Centered ISR Division
Human Trust and Interaction Branch
Wright-Patterson Air Force Base, OH 45433

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
711 HPW/RHXS

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RH-WP-TR-2016-0074

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution A. Approved for public release.

13. SUPPLEMENTARY NOTES
Public Affairs' case number and date cleared: 88ABW-2016-5430, 27 October 2016

14. ABSTRACT
This work addresses automatic transcription for languages without (usable) written resources. Previous work has addressed this problem using entirely unsupervised methodologies. Our approach in contrast investigates the use of linguistic and speaker knowledge which are often available even if text resources are not. We create a framework that benefits from such resources, not assuming orthographic representations and avoiding manual generation of word-level transcriptions. We adapt a universal phone recognizer to the target language and use it to convert audio into a searchable phone string for lexical unit discovery via fuzzy sub-string matching. Linguistic knowledge is used to constrain phone recognition output. Target language speakers are used to assist a linguist in creating phonetic transcriptions for the adaptation of acoustic and language models, by re-speaking more clearly a small portion of the target language audio. We also explore robust features and feature transform through deep auto-encoders for better phone recognition performance. We target iterative learning to improve the system through multiple iterations of user feedback

15. SUBJECT TERMS
lexical discovery, zero resource languages, universal phone recognizer, example-based keyword detection

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
a. REPORT	b. ABSTRACT	c. THIS PAGE			William Kilpatrick
U	U	U	SAR	56	19b. TELEPHONE NUMBER *(Include Area Code)* N/A

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via Air Force Research Laboratory (AFRL). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of ODNI, IARPA, AFRL, or the U.S. government. The U.S. government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright annotation thereon.

1. SUMMARY

The goal of the project was to investigate development of an automatic spoken language processing (ASLP) system without manually transcribed training data. There has been significant progress in the community towards limited-resource and zero-resource approaches for different ASLP tasks.

The work performed under this project lies in the space between zero-resource and limited-resource approaches. It targets cases where no written training resources are available (no lexicon or manual transcriptions), but prior linguistic studies of the language exist and there is some limited access to humans with some level of target language knowledge. Human knowledge of the relationship between acoustic events and linguistic units (phones, words, etc.) is typically reified and transferred to the system in the form of reference transcriptions at the training stage. We aim to acquire such knowledge by alternate, more cost and time efficient means that omit the manual transcription step.

We propose an iterative active learning development process, bootstrapped by input from linguists, during which input from speakers will be elicited via simple audio-based human intelligence tasks (no transcriptions) in order to improve the system. As depicted in Figure 1, unsupervised learning (processes in gray) will be complemented with selective and filtered input from speakers (processes in blue) and linguists (processes in green).

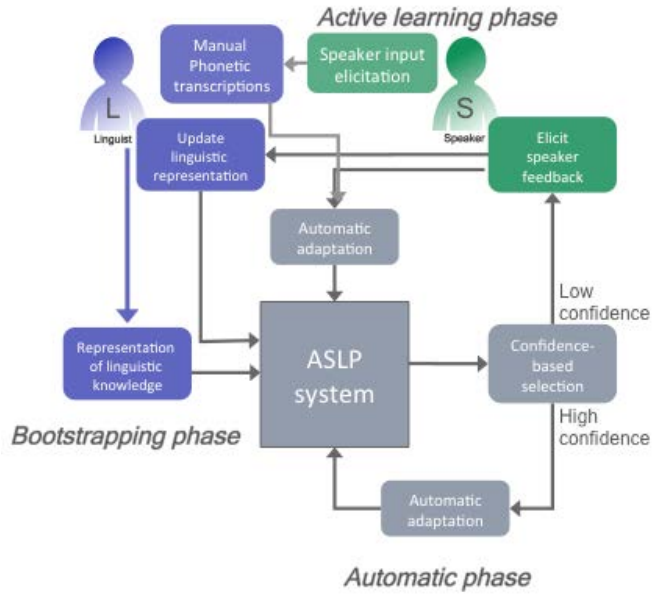

Figure 1: Active learning process for developing ASLP system without use of manual orthographic transcriptions and annotations

The approach is applied to lexical discovery and also evaluated on example-based keyword detection via use of a universal phone recognizer that is iteratively adapted based on linguistic and speaker input. We found that our lexical discovery approach achieves performance comparable to state-of-the-art zero-resource methods that are based on acoustic-only pattern matching, but results in a phonetic based automatically inferred lexicon that can be iteratively adapted and used for follow up applications.

2. INTRODUCTION

State-of-the-art ASLP tasks that involve recognizing words from acoustic input, such as automatic speech recognition and spoken term discovery, typically rely on linguistic resources such as phoneme inventories, pronunciation dictionaries, and annotated speech data. Such resources are unavailable for many languages and are expensive to create.

Recent work on zero-resource spoken language learning has addressed the scenario in which no resources are available for system development ([1], [2], [3]). That research targeted unsupervised phonetic and lexical unit discovery from audio data and produced some interesting results. Nevertheless, these approaches still result in significantly degraded performance compared to using written resources with standardized orthography.

Alternatively, other researchers have concentrated on developing systems using a very small amount of transcribed data, aiming to achieve acceptable performance while minimizing annotation expense. In particular, the IARPA Babel program extended speech recognition capabilities to under-resourced languages, targeting the task of keyword spotting (KWS) in audio ([4], [5]). In the last year the Babel program targeted KWS development with only 3 hours of manual transcriptions – which can be a quick task if native speakers with knowledge of standardized orthography exist. However, even this effort can be very time consuming in the case of languages without standardized orthography, which is the case we target in this work.

Real-world scenarios often lie in the space between zero-resource and limited-resource approaches. Although many languages may lack written resources, they have existing linguistic studies. Typically, researchers and system developers also have access to speakers with some proficiency in the target language, as well as to a wide variety of annotated data in other (possibly somehow related) languages.

This work addresses automatic transcription in languages without written development resources by creating a lexical unit discovery (LUD) framework that benefits from the limited but commonly available linguistic and speaker resources. LUD is implemented as a search of repeated phone strings, by using fuzzy substring matching on the output of a universal phone recognizer (UPR). The accuracy of the UPR on the target language is improved in two ways: (a) by using linguistic knowledge to constrain the recognition output to the target language phonotactics; and (b) by using lightly supervised adaptation on phonetic transcriptions. The phonetic transcriptions are manually created by a trained linguist, but they are facilitated by speakers of the target language who repeat found target language data in a more intelligible manner (re-speaking).

In the following sections we first describe, in section 3, our methodology and algorithms for developing the UPR and LUD framework, and also the target application for example-based KWS. In section 4 we present performance results for the UPR, the LUD and the KWS systems at different stages of system development: (1) baseline, without use of speaker feedback, (2) bootstrap, using linguistic knowledge and limited input from speakers and (3) after some active learning where feedback from speakers is elicited and utilized by the system. We also discuss potential additional improvements of the approach. In section 5 we summarize the conclusions of this study.

3. METHODS, ASSUMPTIONS AND PROCEDURES

The iterative active learning scheme that has formed the basis of the development of the ASLP system developed in the present study is outlined in Figure 1 (shown in section 1.0).

The basic idea is to complement standard unsupervised learning techniques (grey) with input from speakers (green) and linguists (blue). The initial system is bootstrapped with linguistic knowledge represented as finite state transducer (FST) rules that describe characteristics of the language in question. As a critical step in order to improve bootstrap performance, we inserted speaker input elicitation tasks to elicit clean speech (e.g. respeak sentences from available data). This respoken speech can then be used by linguists to create transcriptions with a sufficiently high degree of accuracy. Subsequently, feedback/judgments are elicited from speakers on selected portions of output, via simple audio-based human intelligence tasks that do not require orthography. The portions verified by native speakers are used to update the system (lightly supervised learning). Based on speaker judgments, expert linguists may at that point update the system's linguistic representation. The high confidence portions of the system output can be used for additional unsupervised adaptation. The steps outlined here can be iterated until one of three target conditions are reached:

- speaker feedback suggests no further changes, or
- improvements obtained are only slightly incremental compared to speaker effort, or
- the output is satisfactory for target application.

Two ASLP applications have been addressed in the study:

1. LUD. This application operates on phonetic recognition output and uses acoustic-based pattern matching for bootstrap and as a baseline comparison. A subsequent selection step filters lexical candidates to identify the ones that users could provide reasonable feedback on.

2. KWS. Since we are targeting unwritten languages (no orthographic representation easily available), we focus on **example-based KWS**, where the target keywords are presented to the system in the form of spoken examples (audio files). For the target languages, this is contrasted with dictionary-based KWS, where the target keywords are presented to the system orthographically, and a dictionary pronunciation is found, or derived automatically, based on the orthography and language-specific knowledge. Our KWS implementation was based on phonetic matching and was used to evaluate the quality of the UPR phonetic recognition output.

The underlying enabling technology for the above applications is phonetic-unit recognition, since both LUD and example-based KWS is implemented based on phonetic pattern matching using a phonetic representation of the audio files. We used the UPR system to convert audio files in the target languages to a phonetic string or lattice. In the following subsections we describe our implementation approaches for UPR, LUD and example-based KWS, before we present detailed results for each step of our approach.

3.1 Universal Phone Recognizer (UPR)

3.1.1. UPR Implementation

As mentioned, the target language audio is decoded to a string of phones using a universal phone recognizer. Lexical discovery is then implemented as a search of repeated phone strings, by using fuzzy substring matching on the output of the recognizer, and provides an alternative line of research to methods that work directly on audio.

The universal phone recognizer was created by pooling data from a variety of languages that have written resources. Both the acoustic and language models of our universal recognizer were trained using seven language corpora from different sources: Assamese (Babel), Bengali (Babel), Dari (Transtac), Egyptian Arabic (Callhome), English (Fisher), Mandarin (GALE), and Spanish (Callhome). This gave approximately 650 hours of audio. The data source and amount for each language used is shown in Table 1.

Table 1: UPR training data, languages and data quantities

Language	Source	# hours	#phones in dictionary
Dari	TransTac	126	55
English	Fisher	230	58
Egyptian	CallHome	16	55
Mandarin	GALE	103	57
Spanish	CallHome	19	55
Assamese	BABEL	75	55
Bengali	BABEL	88	55

The acoustic models (AMs) were trained by pooling together the data from all languages. The models were deep neural networks optimized for cross entropy on clustered triphone targets. The front-end features were either mel filterbanks (MFBs) or mel frequency cepstral coefficients (MFCCs).

The language models (LMs) were trained on phonetic transcriptions generated with forced alignments. We used phone bi-grams, as larger n-grams performed worse on unseen (outside the training set) language data. A bi-gram was trained separately for each training language, and the seven individual bi-grams were combined with uniform interpolation.

The dictionaries of the training languages were mapped to a universal phone set. This phone set distinguishes most of the sounds described as phonemically contrastive in the languages, but it also merges acoustically similar sounds that may be contrastive. It does not distinguish pulmonic and non-pulmonic consonants---thus, the ejective consonants of Amharic are not distinguished from their pulmonic counterparts. Consonants with a secondary place of articulation are not distinguished from consonants with only a primary place of articulation---thus, the pharyngealized consonants of Arabic are not distinguished from their plain counterparts. The phone set distinguishes among all manners of articulation, but it merges several contrasts among places of articulation. For example, it does not distinguish the place of articulation between dental, alveolar, and retroflex stops. The set contained 55 speech phones and 3 non-speech

phones. Appendix A shows the inventory of the universal phone set and the phones appearing in each of the training and test languages. In Appendices B and C we show the mappings between the target languages' phone sets (Amharic and Pashto) and that of UPR.

For Amharic phone mapping (see Appendix B) we removed distinctions between:

- pulmonic consonants and ejective consonants, often glottalized in connected speech (9 pairs were merged)
- high mid vowel and high central vowel (the pair was mapped to "ax").

We also mapped labialized phones to sequence of phone with same primary place of articulation and labiovelar glide (acoustically very similar). This reduced phone set drastically, removing about 20 additional phones. Finally, we removed the glottal stops – phonological analyses of Amharic typically include these, though they are not typically pronounced in connected speech. Overall, 31 different phones were either removed or merged with a close one.

For Pashto phone mapping (see Appendix C) we removed distinctions between:

- dental and retroflex stops and nasals.
- retroflex, palato-alveolar, and palatal fricatives (since no dialect distinguishes all three of these places of articulation for fricatives.)
- velar and uvular stops. (uvular stops only in borrowed words from Arabic and not all speakers maintain this distinction.)
- front and back low vowels (acoustically very similar)
- front low, front back, and mid central diphthongs.

Overall about 13 phones were either removed or merged with a close one, while 7 more were mapped to a phone not in the original Pashto phone set.

For Amharic there were more phone merges but less mappings to different phones. For Pashto more phones were found outside the UPR phone set and had to be mapped to a similar phone (7 out of 55), which was probably affecting significantly the performance of the UPR system.

3.1.2. UPR Performance Metric

The UPR performance was measured using phone error rate. However, instead of using the standard phone error measure, we adopted the time-mediated phone error rate (TPER) variant, which is better at aligning references and hypotheses in a linguistically meaningful way. As is illustrated in the following example, simple string comparison produces unlikely confusion pairs such as `aw/s`, `x/ae`, `ax/l`, and `s/ax`:

```
REF:  aw x  ax z aj d e b a s
HYP:  s  ae l  x n  S i b * ax
```

By taking time information into account TPER instead yields more plausible confusion pairs such as `aw/ae` and `a/ax` in this situation:

```
REF:  * aw * x ax z aj d e b a  s
HYP:  s ae l x ** * n  S i b ax *
```

As a result of choosing TPER for measuring performance, the error rates reported here are 3-8% higher than they would have been for standard phone error rate.

3.2 Lexical unit discovery

3.2.1. Lexical unit discovery algorithm

We developed an automatic lexical unit discovery approach to discover repeated words and phrases in continuous speech, based on output from the universal phone recognizer on raw speech data for a target language. The approach takes the recognized phone sequence as input and discovers lexical units by finding repeated substrings. We extended the standard repeated substring detection algorithm with two modifications. First, we used the silence labels in the phone recognition output and the sentence boundaries of the audio segments to be recognized as boundaries of lexical units. Second, the standard substring matching algorithm was extended to fuzzy matching by using the substitutable, deletable, and insertable phones for the target language, as defined by linguistic knowledge. We used this extension to model variability (i.e., different phone sequences correspond to the same word).

3.2.2. Related Work

Related work to our approach of identifying lexical units from unsegmented strings of symbols includes ([7], [8], [9], [10]). Some prior work models word segmentation based on the distribution of phoneme sequences in the input (e.g., [9]). Some prior work explores synergistic interactions between multiple levels of linguistic structures (e.g., [8]). In this work, we also investigated the effectiveness of our robust features and features transformed by deep auto-encoders for lexical unit discovery using the minimum dynamic time warping (DTW) alignment cost DTW(w_i, w_j) for a word pair (w_i, w_j), as proposed in ([6]). As the baseline approach in bootstrap phase and a related work to compare to our approach, we used the minimum DTW alignment cost implementation in the Zero-Resource Speech Discovery, Search, and Evaluation Tools (https://github.com/arenjansen/ZRTools), and used cosine distance as the distance metric between constituent frames in the word examples.

3.2.3. LUD Performance Evaluation Approach

To evaluate LUD performance, we adopted the evaluation approach proposed in ([6]), which provides a rapid measurement of how well a speech representation can associate examples of the same word types and discriminate different word types, as well as to assess speaker independence of the speech representation. Using the time-aligned word reference transcripts of a dataset of the target language, we randomly sampled a set of spoken word examples with a minimum character length of 5 in the surface form, denoted $W=\{w_i, i=1,\ldots,M\}$. We then randomly sampled the following four sets of word pairs (w_i, w_j) in{WxW}, i is different from j) from the dataset ([6]):

- C1: Same word, same speaker (SWSP)
- C2: Same word, different speakers (SWDP)
- C3: Different word, same speaker (DWSP)
- C4: Different word, different speakers (DWDP)

For the target languages of Amharic and Pashto, we used the development sets from the Babel program to evaluate the lexical unit discovery performance. Table 2 shows the word counts and

sizes of the four sets (C1, C2, C3, C4) sampled from the time-aligned word transcripts of these two datasets

Table 2: Word counts and sizes of the four sets (C1, C2, C3, C4) for lexical unit discovery evaluation

Dataset	Word Count	|C1|	|C2|	|C3|	|C4|
Amharic	47,292	3,679	123,763	386,150	1,016,263
Pashto	94,721	3,567	100,939	308,632	1,037,561

On these subsets of the target language, we first ran phone recognition and then automatic lexical unit discovery. For each word pair (w_i, w_j) in the four sets, we assigned a discovered repeated phone sequence (i.e., discovered lexical unit) Sw_i to w_i, and a discovered lexical unit Sw_j to w_j, if the discovered lexical unit Sw_i and Sw_j overlap at least 50% of the duration of w_i and w_j, respectively. Then the distance of (w_i, w_j) was computed as the Levenshtein distance between the two phone sequences of lexical units Sw_i and Sw_j, divided by the maximum of the lengths of the two phone strings hence converted to a value in [0,1]. If multiple discovered lexical units overlap the duration of w_i or w_j, the distance was computed as the smallest distance between the discovered lexical units. If there is no overlapping discovered lexical unit to w_i or w_j, then we assigned a distance of infinity for the word pair (w_i, w_j). Then, given a threshold tau, we can compute

$$N_k(\tau) := |(w_i, w_j) \in C_k : Distance(w_i, w_j) \leq \tau| \qquad (1)$$

and then compute the precision recall for SW, SWSP, and SWDP as follows:

$$P_{SW}(\tau) = \frac{N_1(\tau) + N_2(\tau)}{\sum_{k=1}^{4} N_k(\tau)} \qquad (2)$$

$$R_{SW}(\tau) = \frac{N_1(\tau) + N_2(\tau)}{|C_1| + |C_2|} \qquad (3)$$

$$R_{SWSP}(\tau) = \frac{N_1(\tau)}{|C_1|} \qquad (4)$$

$$R_{SWDP}(\tau) = \frac{N_2(\tau)}{|C_2|} \qquad (5)$$

We sampled through a series of values of tau and computed the Precision-Recall Breakeven (PRB) point, PRB_{SP}, where P_{SW} and R_{SWSP} are equal, and PRB_{DP}, where P_{SW} and R_{SWDP} are equal. Note that a high PRB_{SP} value indicates a good speaker-dependent speech representation for spoken term discovery, and a high PRB_{DP} value indicates a good speaker-independent speech representation for this task.

3.3 Example-based KWS

In order to assess the performance of our phone recognizer, we developed a keyword detection system where the keywords are provided to the system through examples.

The architecture of the example-based KWS system is shown in Figure 2.

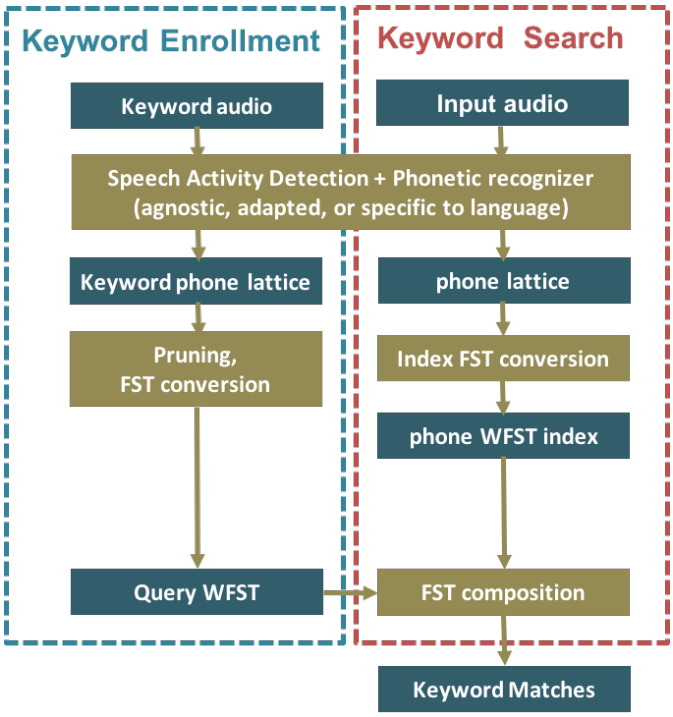

Figure 2: Example-based KWS system architecture

First, an example recording is made of each keyword; this process is called **keyword enrollment** (similar to speaker enrollment in speaker identification systems). After using Speech Activity Detection (SAD) to strip out the noise and silence from the example recording, we run our phone recognizer to obtain a phone lattice for each keyword which encodes all possible pronunciations. Then the lattice is pruned, to keep only the top pronunciations, and an FST acceptor is created which allows for fast and efficient search. The beam that is used to prune the keyword lattices was tuned on the dev data. A larger beam can improve recall by including more pronunciations, but can also create spurious detections since those pronunciations are less reliable.

On the search data, the process of **keyword search** involves the following steps: First we perform SAD and then create a phonetic lattice for each audio recording, which is then converted to a weighted phonetic index. That index is searched with the query FST to create the resulting keyword detections. Those detections are then calibrated to improve the score quality and help re-rank detections from different keywords. The calibration uses posterior and keyword-rank as features and logistic regression for re-scoring, as described in [12].

The main difference between example-based and dictionary-based keyword detection is the enrollment step via audio examples. For dictionary-based, which has been the setup used in other programs (including Babel), the keywords are presented to the system via their orthographic representation, and the system uses a dictionary pronunciation (provided or automatically derived) to create the representation (in our case the FST acceptor) that is used in the search state. We used the dictionary-based search (with provided keyword pronunciations) for comparison purposes, as this provides an upper-bound of performance for our approach.

4. RESULTS AND DISCUSSION

Our results section is organized per stage in the iterative learning approach. We start by presenting the baselines (initial unadapted results) and then proceed to results obtained from the bootstrap stage and after feedback elicitation (1 iteration) from speakers. We did not obtain any results with unsupervised adaptation or with additional iterations from speakers.

4.1 Baseline results

4.1.1. UPR initial performance

We evaluate our results on the Amharic and Pashto development sets from the IARPA Babel program. This audio was collected under a variety of real-world conditions and contains background noise. Both development sets had 10 hours of audio available. After using the utterance segmentations from Appen (marking the sentences and removing silences), the Amharic testset has approximately 7 hours and contained 50,000 words, and Pashto has 8 hours and 100,000 words. Lexica and transcriptions are available for these languages, and these were used to evaluate our experimental systems that do not rely on such resources.

We scored the phonetic recognition output against phonetic references generated from a forced alignment and computed TPER results. The results of both languages were analyzed and optimized during development, so all results should be viewed as development set results.

The results in Figure 3 demonstrate the effectiveness of different sets of training data on unadapted UPR performance. All results use acoustic models based on 40 mel-scaled filterbanks (MFBs). First we report the results for our original set of training languages: Dari, English, Egyptian, Mandarin, Spanish (labeled "DEEMS"). Then we show the effect of adding Assamese and Bengali to the training pool (labeled "+AB"). Finally, we compare with the result when trained on the full Babel target language training sets as a limiting case (labeled language specific). For language specific training, we used all the Babel provided data for the target languages. Amharic had about 40 hours of conversational and 14 hours scripted audio and Pashto about 80 hours of conversational and 34 hours scripted. For the conversation portions we only used the audio in the Appen transcribed segments (removing about 20% of the audio). Overall we had about 42 hours of segmented training audio for Amharic and 96 for Pashto.
training audio for Amharic and 96 for Pashto.

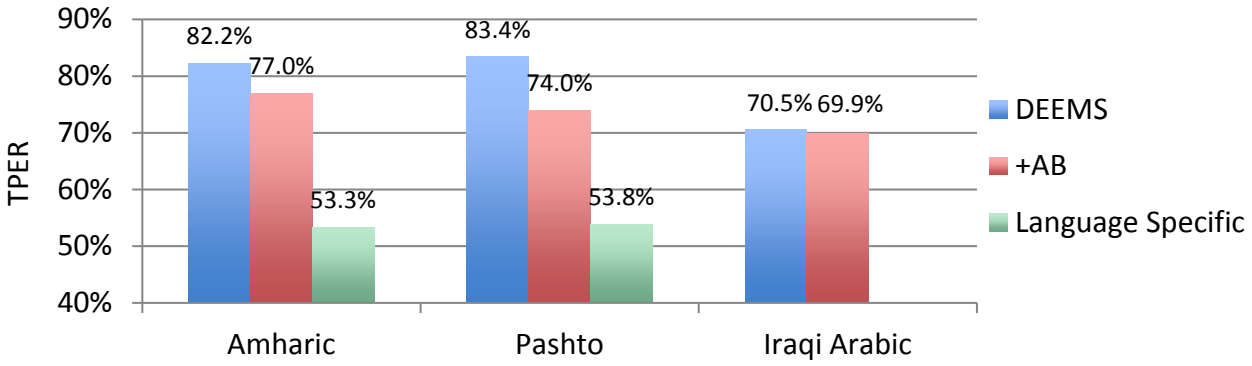

Figure 3: Initial UPR performance on target languages and Iraqi Arabic (additional language). Results are reported using the DEEMS (Dari, English, Egyptian, Mandarin, Spanish) and the +AB (DEEMS +Assamese+Bengali) training data. Both are compared to language specific (Amharic and Pashto) trained models.

We observe from Figure 3 that the addition of the "+AB" datasets had a significant impact in both Amharic and Pashto. We selected these languages (Assamese and Bengali) because of their acoustic similarity to the target languages, especially to Pashto, which probably explains why the improvement for Pashto is larger than for Amharic. We also show the results on an Iraqi Arabic testset from the Transtac program. The TPER results on that set are significantly better than for Amharic and Pashto, since that is a much easier set (broadband, clear audio recordings). In addition, Iraqi Arabic is acoustically similar to two of the original DEEMS languages (Dari and Egyptian) and does not stand to gain from the additional "+AB" sets (the figure shows no significant difference for Iraqi between DEEMS and +AB).

The results on the Iraqi testset were merely presented at this point to demonstrate the effect of choosing the appropriate languages to add as training data for the UPR. If there is not sufficient similarity/coverage of the target language, then adding extra languages makes a big difference. However, unsupervised adaptation on the target language can cover some of this difference, as we will show next.

In Figure 4 we demonstrate the effect of unsupervised adaptation using feature-space maximum likelihood linear regression (fMLLR). The front-end for all fMLLR results was 13 mel-frequency cepstral coefficient (MFCC) features along with deltas and double deltas. We observe that unsupervised adaptation using fMLLR gives significant improvements across both testsets. Its impact was larger for the DEEMS training set than for the full +AB set – which shows that adaptation covers some of the benefit we can get from adding extra training data from similar languages. Still the +AB system was still significantly better after fMLLR unsupervised adaptation than the DEEMS system for both languages

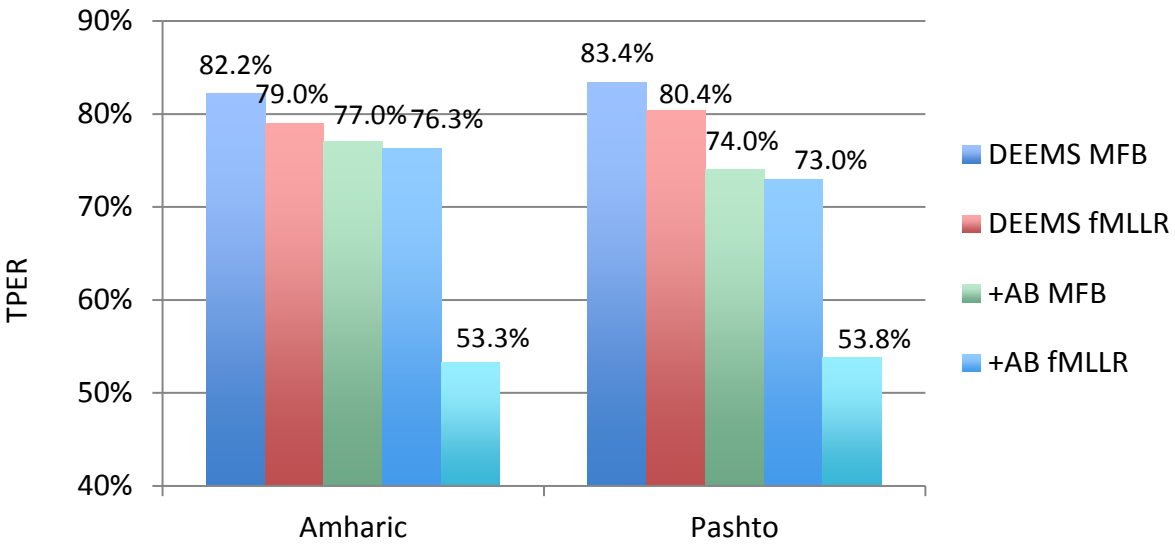

Figure 4: Effect of unsupervised adaptation on UPR performance

Next, we explored the effect of including the silence phone in the LM. Typically, silence is not included in a word-based LM, and silences are optionally inserted between words using a probability that does not depend on the surrounding words. In our work, we decided early on that it made sense to include silence in the phone LM, since it gave some preliminary better results –

at that time our TPER was higher than 90%. In Figure 5 we show the effect of removing silence from the LM in the best result we had at Figure 4 (using the +AB fMLLR AM). As expected, including silence in the language mode (LM) gives significantly better results. Silence is somewhat modeling word boundaries and it is very useful to include as part of the phone ngram. Moreover, allowing silence to be inserted with no penalty at any time results in high phone deletion rate. Thus, including silence in the LM was a good choice and it is included everywhere in our reported results in this work.

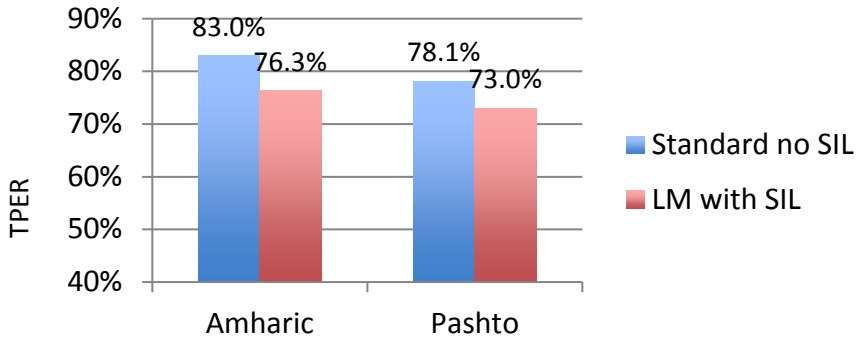

Figure 5: Effect of including silence label in the UPR LM

4.1.2. Baseline Lexical Unit Discovery Results

In Table 3 we show the LUD results both from our approach and from the DTW-based speech pattern discovery, from Jansen's ZRTools ([6]). We present the PRB_{SP} and PRB_{DP} values on the Amharic and Pashto development sets.

For the UPR-based we used the output from the un-adapted baseline UPR (using the +AB MFB AM and un-adapted LM), and compared with the Target-Language phone recognizers (TLPRs) for both Amharic and Pastho (denoted as Language Specific on the table). The time-mediated phone recognition error rate (TPER) for each output is also shown along with the PRB_{SP} and PRB_{DP} results. We observe that PRB_{SP} and PRB_{DP} scores are correlated with the phone recognition accuracy, i.e., with improved time-mediated phone recognition error rate, these scores generally got improved.

Table 3: Results from lexical unit discovery evaluation from our phonetic lexical unit discovery approach with the baseline system (no adaptation) and from using various features, including our novel features, in Jansen's ZRTools

	Amharic			Pashto		
	TPER (%)	PRB_{SP}	PRB_{DP}	TPER(%)	PRB_{SP}	PRB_{DP}
UPR-based Phonetic Approach						
Baseline	77.0	0.19	0.19	74.0	0.18	0.18
+ boundary + silence info	77.0	0.23	0.23	74.0	0.22	0.22
+ ling. Phone confusions		**0.25**	**0.25**		**0.23**	**0.23**
Language Specific (upper bound)	53.3	0.43	0.43	53.8	0.42	0.42
Jansen's acoustic-based Approach						
MFCC feature	N/A	0.33	0.14	N/A	0.35	0.16
NMC (SRI's noise robust) feature	N/A	0.47	0.40	N/A	0.48	0.40
BN-DAE (SRI's eutoencoder)	N/A	0.42	0.38	N/A	0.46	0.39

We analyzed the efficacy of using sentence boundary and silence labels as boundary constraints for lexical unit discovery, as well as the efficacy of using linguistically defined confusable phones for fuzzy matching. As shown in the table, using the boundary constraints made a significant improvement on the lexical unit discovery performance, whereas using the linguistically defined confusable phones for fuzzy matching hasn't been able to make a significant impact.

In Table 3 we can also see how the Jansen's LUD approach is affected by the use of different features as input to the ZRTools. The raw MFCC features are compared with our robust normalized modulation coefficient (NMC) features and the features transformed through deep auto-encoders (denoted BN-DAE). We observed a significant improvement on the PRB_{SP} and PRB_{DP} scores from our robust NMC features and BN-DAE features, especially a significant improvement on cross-speaker performance.

The best result (with NMC features) from Jansen's approach is comparable with the Language-Specific result for the phonetic-based approach (for within speakers' results it is even better). Thus we cannot hope to improve over that with our UPR approach – even if we improve the UPR performance to the level of TLPR. But we observe that the LUD approach based on phone recognition output shows strong speaker independence as PRB_{DP} is the same as PRB_{SP}, for both target languages. This is very important, in the case when limited data can be found for the same speaker.

Apart from speaker independence performance, the other advantage of the phonetic-based LUD is that the output results can be immediately readable and interpretable by linguists. The phonetic representation is a standard way of storing lexical units in dictionaries that most ASLP systems utilize (e.g. speech recognition, KWS, information extraction, topic modeling). The output from Jansen's tool is an acoustic representation, so it is not easy to use for dictionary generation nor use for follow up ASLP applications.

Moreover, standard acoustic model adaptation approaches can benefit phonetic based models, so our plan is to improve the performance of the UPR-based system by providing some adaptation data through iterative user-feedback elicitation. We also can explore other modeling approaches, e.g., noisy-channel model ([3]), to learn models for variations for our phonetic approach – which was not within the scope of this study.

4.1.3. Baseline KWS Results

We compare the performance of the two types of KWS systems (example-based and dictionary-based) with the Target-Language phone recognizer (TLPR, trained on Amharic data only) and the Universal phone recognizer (UPR, trained on multilingual data without Amharic). Experiments were run in the Amharic development set, using a set of 1289 keywords, which were a random subset of the official dev Amharic Babel keywords for which we gathered spoken recordings (it was too timely expensive to gather spoken recordings for all the keywords in the development setup for Babel).

Figure 6 shows detection error tradeoff (DET) curves for the pronunciation-based and the example-based KWS systems. It is clear that, when the phonetic system is trained on matched Amharic data the dictionary-based keyword search (solid black line) produces better results than

use of recorded examples for keywords (dashed black line). But, when we only have a weaker phone recognizer available (the using the UPR output), the example-based enrollment (dashed blue line) is beneficial over a dictionary-based system (solid bleu line). That is true even considering the fact that the examples for the keywords were recorded in clean environment, which is different from the Babel audio conditions, and we believe it is due to the fact that the UPR output is not a good representation of the dictionary pronunciations, but it describes somewhat consistently the keywords both in the examples and in the searched audio.

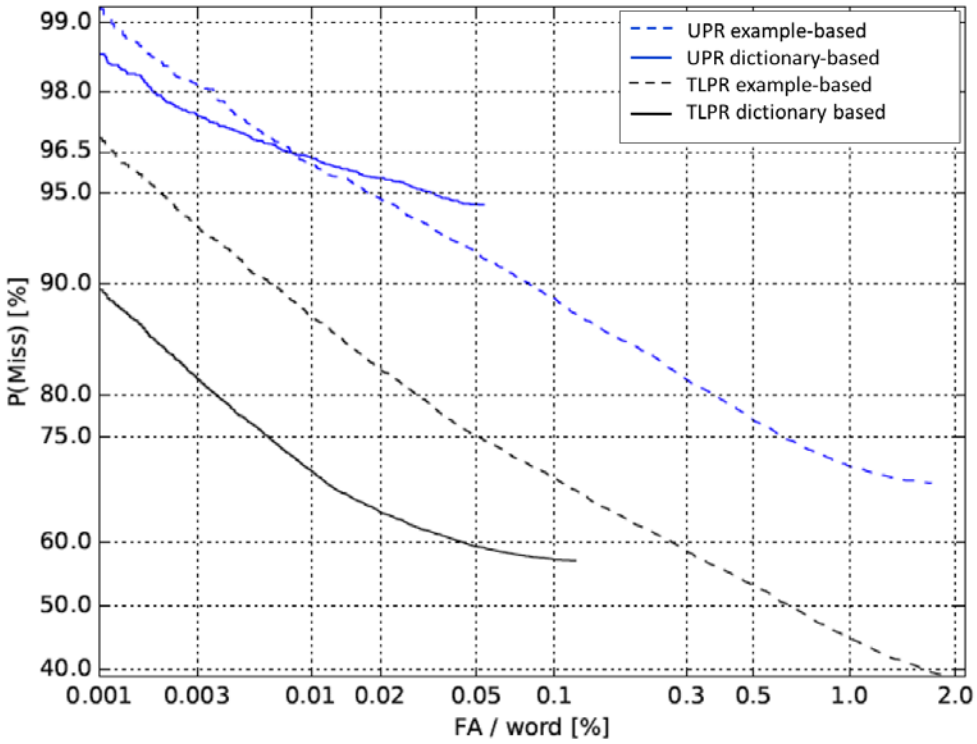

Figure 6: Comparison of KWS results using baseline UPR system and TLPR, both with example-based and dictionary-based keyword enrollment.

4.2 Bootstrap Step

The bootstrap step, as implemented in the current study, consisted of three subtasks: (1) significant facts about the target languages were captured by means of linguistic constraint language. (2) native speaker input was used to respeak sentences from the available corpora which were phonetically transcribed by trained linguists and (3) the respoken transcribed data was used to adapt the UPR system and improve performance.

4.2.1. Linguistic Constraints

The goal of the initial phase of the boostrap step is to capture language constraints in terms of possible syllables (and syllable sequences) in a word. In the field of linguistics, such constraints are known as the "phonotactics" of a language. Languages differ in the degree to which they constrain possible syllables, so the impact of these linguistic constraints to some extent depends on the strength of the phonotactic constraints. For instance, a language that exclusively permits CV syllables constrains permissible phone sequences much more than one that permits closed syllables and complex onsets and codas. Other knowledge can be brought to bear as well, such as

known affixes or stem morphemes; however, this kind of knowledge is hard to operationalize if nothing more is known about the target language utterance.

Constraints on possible phone sequences were captured by virtue of the well-established Xerox finite state tool (XFST) rule format and compiled using the open-source Foma rule compiler ([11]). Foma also allows the random generation of successful traversals through the network, which can be used to create artificial data. The constraints were compiled into finite state transducers, and the recognizer's transducer was composed with the constraint transducer to limit its output.

The number of rules that can be devised depends on the information that can be found easily in linguistic studies for the target language. Studies typically describe the language's phonetic inventory, and this approach allows us to easily take advantage of this by constraining the output of the recognizer to the target language's phone set. Decoding can be further constrained by taking the language's syllable structure into account. Amharic has a particularly well-defined syllable structure. It allows no more than one consonant in the syllable onset position (except labialized consonants), and no more than two consonants in other positions (e.g., (C)V(C)(C)). Pashto allows for complex onsets consisting of up to three consonants, and complex codas of up to two consonants (e.g., (C)(C)(C)V(C)(C)). The sequence of consonants in Pashto's onsets and codas is further constrained, in part, by the sonority hierarchy and other similar constraints. Even so, Pashto allows for a greater variety of syllables than Amharic.

A schematic rule set for Amharic is given below. It defines possible phone strings as sequences of one or more phonetic words flanked by optional silences. Words in turn are non-zero sequences of syllables, which in turn can be closed or open. Finally, the "V" and "C" classes define the vowel and consonant inventory of the language.

```
define PhoneString ({ SIL }) Word+ ({ SIL });
define Word Syllable+;
define Syllable ClosedSyllable | OpenSyllable;
define ClosedSyllable Onset Nucleus Coda;
define OpenSyllable Onset Nucleus;
define Onset SimpleOnset | ComplexOnset;
define SimpleOnset [ C - { 44 }];
define ComplexOnset ...;
…
define Nucleus V;
define V [ { a }|{ e }|{ i }|{ o }|{ u }|{ ax }|{ 1 }];
define C [ { 4 }| … ];
…
```

The complete sets of Foma rules we developed for Amharic and Pashto are shown in Appendices C and D respectively. There was a total of 17 "define" expressions for each of the two languages. If a good phonological description is available, an initial rule set for a language can be produced within a day, with the possibility of further refinements as more facts about the language get integrated.

The impact of using linguistic constraints to TPER performance is shown in Table 4 below, where `Phone` shows the effect of limiting the UPR output to the language-specific phone set (could be defined as a single Foma rule) and `Syllable` shows the effect of constraining in

addition the UPR output to be compliant with the basic syllable structure in each language (full set of rules).

Table 4: Effect of including phone-based and syllable-based constraints

Constraint	Amharic TPER(%)	Pashto TPER(%)
None	76.3%	73.0%
Phone	75.1%	71.8%
Syllable	74.6%	71.5%

We observe from the table that the maximum benefit for UPR is obtained by just constraining the inventory to the phones of the target language. Although this improvement is straightforward, it demonstrates how this commonly available knowledge can be utilized by our approach. The syllable constraint gives additional small improvement for Amharic. The improvement for Pashto is smaller, as Pashto has more syllable variety than Amharic. It is possible that because the syllable constraints were applied on top of the universal phone bigram, which may not be very favorable to some of the constraints, we can only see a very small improvement from the constraints at this stage. As we will see in the adaptation section, there is a bigger relative benefit observed if the recognizer's bigram is first adapted on target language data.

4.2.2. "Respeak" Speaker Task

The best results so far for the UPR (in table 4) are still pretty low to allow for successful lexical discovery. Thus it was necessary to obtain a (small) amount of adaptation data in order to improve the results. Since we are targeting unwritten languages, we did not want to assume any knowledge of orthography, thus the only way to transcribe the data would have to be at the phonetic level. However, the narrow-band conversational audio proved too hard/costly/error-prone for linguists (i.e., non-experts of target language) to transcribe phonetically.

Therefore, we devised a first task for the target language native speaker(s) in which they were asked to respeak a small amount of original data in three modes: (1) same speech rate, (2) reduced speech rate, and (3) same rate with explicit pauses inserted between words. The illustration in Figure 7 below shows a screenshot of the tool implemented for the "Repeat Word" task, as it was used by one of our Amharic native speakers.

Originally, we randomly selected 50 utterances (about 4 minutes total) from the segmented Babel training corpus to present to speakers for the Respeak task. In real life we don't expect to have segmentations but we will apply SAD to preprocess the original audio. It is important not to have too long segments (less than 10 seconds) since longer sentences are harder to remember exactly and make the Respeak task less reliable.

The speakers listened to each utterance and were asked to Respeak it in the three modes described above. They were instructed to skip the utterance if the audio segment contained no speech, speech outside the target language, or was unintelligible.

For Amharic, out of the 50 presented, 34 utterances were respoken and transcribed (about 3 minutes). For Pashto, to target a larger amount, we presented ~100 utterances to 2 different speakers, and ended up with 92 respoken utterances, which were all transcribed.

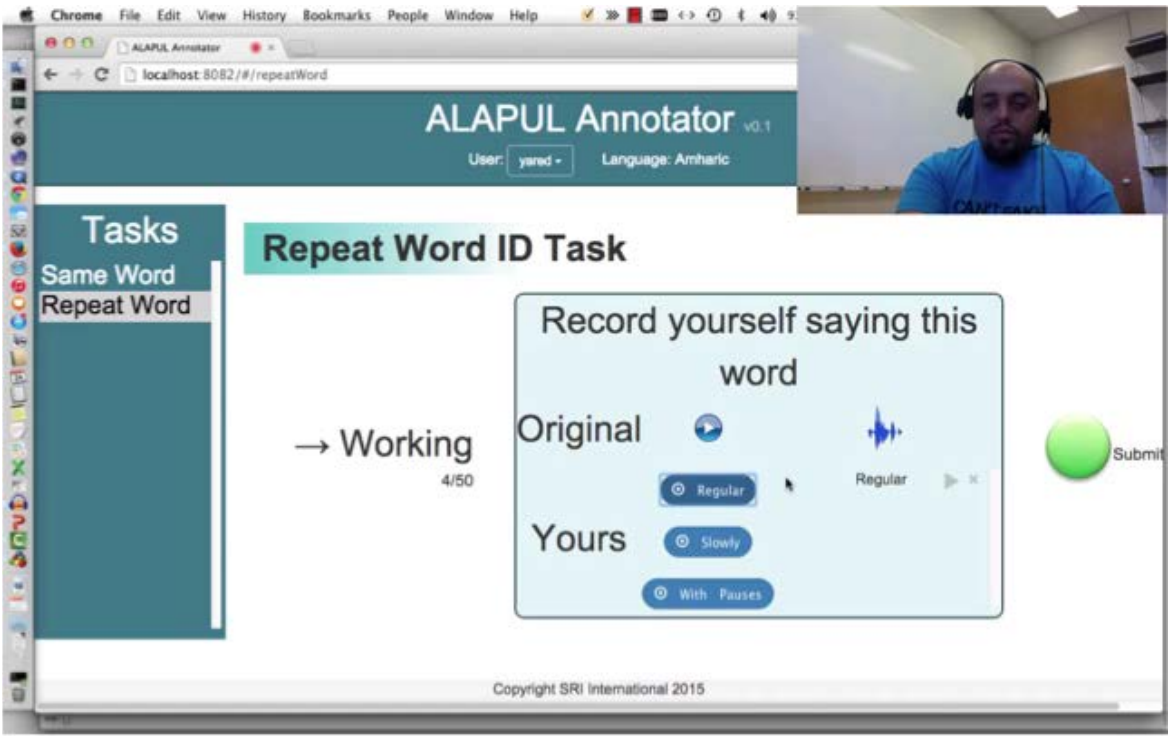

Figure 7: Screen capture of RespeatWord task

4.2.3. Respoken Data Transcription

The newly obtained broad-band recordings, in particular the versions with slower speaking rate and/or pauses, proved essential in creating reliable transcriptions for adaptation purposes. Using the original narrow-bandwidth audio, a random set of 100 short (4-16 words) Pashto utterances, took about 3 minutes per utterance, i.e., ~54x (real time), to transcribe and was quite error-prone. Working with respoken data, we were able to provide transcriptions at the speed of about 25x (real time), with a much greater degree of accuracy. In total, it took less than 8 hours of linguist labor per ~100 utterances.

The linguist first listened to all the respoken versions (`Matched, Slow, With-Pauses`), and then transcribed the matched and slow versions. The `Slow` and `With-Pauses` versions were used to inform the transcription of the `Matched` audio, but many differences remained due to phonological reduction and assimilation processes. The transcriptions were compared to ensure that any differences were truly reflected in the pronunciation and not due to transcription variability. The audio was initially transcribed using a narrow phone set designed for UPR (based on the International Phonetic Alphabet). The transcribed phones were then mapped to a set of phones being used to represent the target languages. Any phones outside of this set were mapped to closest in-language phone, using the mappings provided in Appendices B and C.

4.2.4. UPR Supervised Adaptation at Bootstrap step

The small amount of phonetically hand-transcribed broadband Amharic and Pashto sentences described in section 4.2.2, were used for adapting both the AM and the LM of the UPR. The Deep Neural Network (DNN) AM was retrained in its entirety without changing the network structure, using small steps for parameter updates. For the LM we explored maximum a-

posteriori (MAP) adaptation as well as the linear interpolation of the background language model and a small LM trained on the adaptation data. Linear interpolation showed much better perplexity results on the target language test sets than MAP adaptation.

The results for adapting the recognizer are given in Table 5. The `Hand,Babel-only` line corresponds to an AM adapted using the original Babel audio together with the manually-generated transcriptions. The `Hand,All` result uses for adaptation both the original Babel audio as well as the respoken audio (from the `Matched` and `With-Pauses` recordings) by our native speaker. Overall, for the `Hand,All,` we had 97 Amharic utterances and 265 Pashto utterances available for adaptation (a few of the respoken modes were not recorded properly for some utterances thus we don't always have 3 times the original Babel data). The combined set gave us additional improvement, compared to using the Babel only utterances.

The use of LM adaptation is shown in the `LM Adapt` column of Table 5. `Hand` denotes adaptation with the hand-generated phonetic transcriptions. Additionally, we applied the syllable constraints defined in the Foma grammars for each language (as described in section 4.2.1) to further constrain the recognition output to the known language phonotactics. This is denoted as `+syl constr`, when used in the `LM Adapt` column.

Finally, to show the potential effect of a larger transcription effort we also give results for adaptation using 30 minutes of audio with forced alignment transcriptions, labeled `Forced-30'` in Table 5. 30 minutes correspond to 450 utterances for Amharic and 385 for Pashto. As an upper bound for the potential of adaptation, the table has results for models trained using all target language training data, labeled `Language Specific`. In that case, forced alignments from target language transcriptions were used for both the acoustic and language models.

Table 5: Recognizer adaptation results.

AM Adapt	LM Adapt	Amharic TPER(%)	Pashto TPER(%)
Unsupervised (baseline)	None	76.3%	73.0%
Hand,Babel-only #utterances: 34 Amharic 92 Pashto	None	71.8%	71.4%
Hand,All #utterances: 97 Amharic 265 Pashto	None	70.0%	71.5%
Hand,All #utterances: 97 Amharic 265 Pashto	Hand #utterances: 34 Amharic 92 Pashto	67.9%	70.7%
Hand,All #utterances: 97 Amharic 265 Pashto	**Hand+syl constr.**	67.2%	70.6%
Forced-30' (comparison) #utterances: 450 Amharic 385 Pashto	Forced-30'+syl constr	62.0%	64.5%
Language Specific (upper bound)	Language specific	53.3%	53.8%

As we observe clearly from Table 5, AM adaptation, even with very small amount of adaptation data, significantly helps TPER. The relative improvement for Amharic is bigger than Pashto, even though we used more adaptation utterances for Pashto (3 times more). For both languages there is gain from additionally applying LM adaptation (again larger gain for Amharic). The additional improvement using the `syl constr` is consistent with the results in Table 4. We see a bigger improvement for Amharic, since as we mentioned earlier the Pashto syllable structure is more variable.

The larger gain for Amharic for adaptation is probably due to the fact that the Pashto data has more dialect variation. For the collected Respoken data, we had 2 speakers from 2 regions (north and south) and instructed them to try to respeak the data in the dialect they heard it. When, at an earlier stage, we used only one speaker (and only similar amount of data to Amharic) we saw almost no improvement, so either the first speaker had a significant accent difference from the target testset (which our linguist didn't think was the case), or data from a single speaker was not enough to cover the variability in the data. Data from two speakers definitely achieved some improvements for adaptation in Pashto, but even the 30-minute comparison shows smaller improvements for Pashto than for Amharic. For Amharic, we seem to be getting most of the improvement from adaptation from the first 100 utterances (`Hand,All` - which includes the respoken data) while using the 30 minutes only improves a smaller relative amount. For Pashto we see the opposite trend: we get little improvement using 100 or so original Babel data with transcriptions (`Hand,Babel-only`), no additional gain from the respoken data (`Hand,All`), so the speakers we are using may indeed not match very well the data, while a larger chunk of improvement can be achieved with the 30-minute adaptation data. So it seems that indeed much more data is needed for Pashto to achieve similar amount of improvement as in Amharic, which indicates the data is more variable. In addition, as we noted in section 3.1.1, for Pashto there were more phones not found in the UPR phonetic inventory, and we had to map these phones to similar phones in the inventory. That may be affecting adaptation, since more data may be needed for those phones to make the AM a match to the target language observed data.

4.2.5. Lexical Unit Discovery After Bootstrap Adaptation

Table 6 below shows the LUD performance based on the output from the phone recognizers adapted on the manually created phonetic transcriptions from the bootstrapping phase, and the phone recognizers trained on the training set of the target languages, respectively. We include the best result for the Jansen's acoustic-based approach from Table 3.

Table 6: Results of our phonetic lexical unit discovery from the best bootstrap adapted UPR system

	Amharic			Pashto		
	TPER(%)	PRB_{SP}	PRB_{DP}	TPER(%)	PRB_{SP}	PRB_{DP}
Baseline + boundary constr.	77.0	0.23	0.23	74.0	0.22	0.22
+ phone confusions		0.25	0.25		0.23	0.23
+ Bootstrap Adaptation	67.2	0.36	0.36	70.6	0.31	0.31
+ syl constraints		0.40	0.40		0.37	0.37
Jansen's with NMC feature	N/A	0.47	0.40	N/A	0.48	0.40
Language specific (upper bound)	53.3	0.43	0.43	53.8	0.42	0.42

The best bootstrap adapted UPR achieved 67.2% TPER on the Amharic dev set and 70.6% on the Pashto dev set. We consistently observed that our LUD approach based on phone recognition output shows strong speaker independence as PRB_{DP} is the same as PRB_{SP}, for both target languages. We also observed that PRB_{SP} and PRB_{DP} scores are correlated with the phone recognition accuracy, i.e., with improved TPER, these scores got improved. It is encouraging to see that the LUD performance from the best bootstrap adapted phone recognizers is close to the performance from the TLPR systems.

One important point from Table 6 is that we get significant improvement of the LUD performance when we apply the syllable constraints. Even though these constraints were already applied to constrain the output of the UPR, we get an additional improvement when applying them to constrain the LUD thus making sure that our discovered units are consistent with the language phonotactics. This observation is true on both languages.

The second point we want to make is that our current best result is now very close (at least in cross-speaker discovery) to the best result we obtained using the acoustic-based approach from Jansen's tool and our robust acoustic features. This is very important since it shows that the bootstrap adaptation provides a good starting point for selecting candidates for user feedback.

4.2.6. KWS after bootstrap adaptation

In this section, we present the KWS results in Amharic using the adapted UPR and compare with that of the baseline UPR and the TLPR. As in section 4.1.3, we present both the example-based results along with dictionary-based. Figure 8 shows the DET curves of these results.

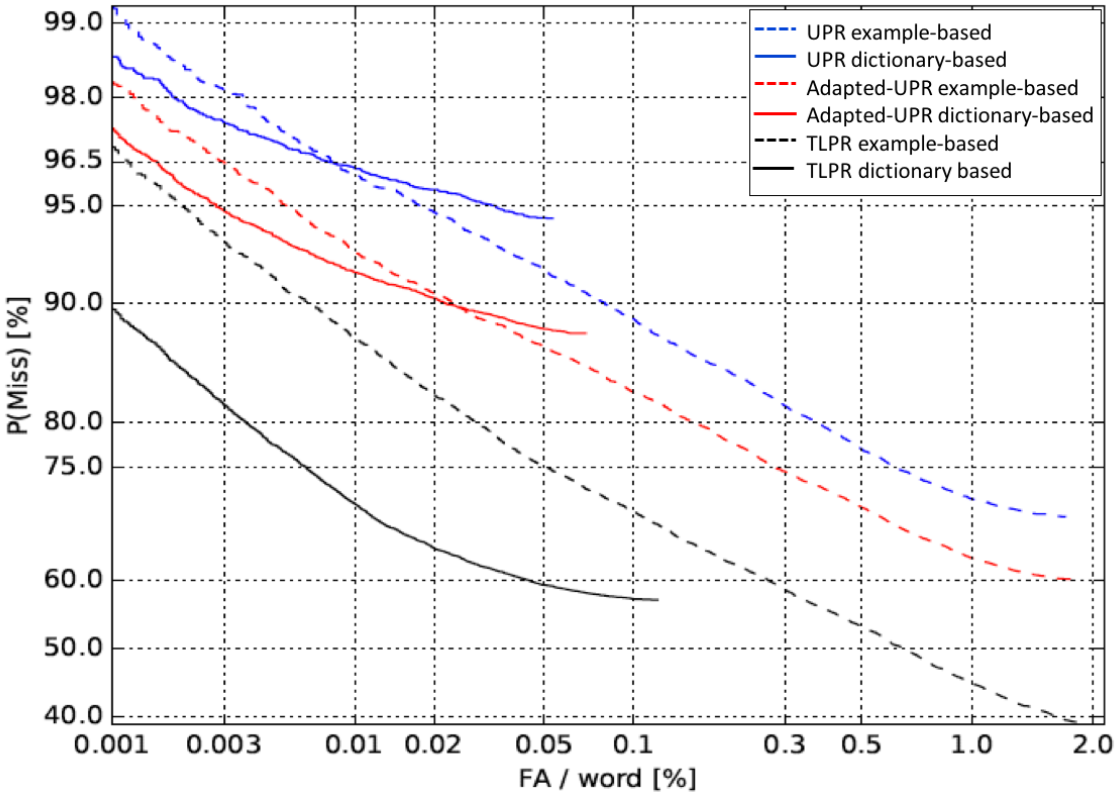

Figure 8: Comparison of KWS results for Amharic using baseline UPR, TLPR, and adapted UPR, both with example-based and dictionary-based keyword enrollment

In Figure 8, we find that UPR adaptation to target language bridges part of the gap between un-adapted UPR (baseline) and TLPR. It is interesting that the improvement in TPER performance corresponds to almost identical relative improvement in KWS.

As we also observed for the un-adapted UPR, example-based keyword enrollment enables higher recall and precision than dictionary-based, when looking at high-recall operating points. If looking at higher-precision operating points, dictionary-based enrollment seems to always outperform example-based. This behavior is due to the fact that in the example-based setup, the highest scoring detections (where the system's phone recognition has a high confidence) originate from several possible pronunciations while in the dictionary-based KWS they originate from the one correct pronunciation, thus making the latter less prone to false alarms.

The community is used to report Actual Term Weighted Value (ATWV) [13] or the Maximum Term Weighted Value (MTWV) for KWS performance. In our case only about half of the Amharic keywords had spoken examples, so we restricted our evaluation to those keywords in order to have a fair comparison between techniques. We only computed MTWV scores, since it was hard to tune for the optimal value (needed a separate set). For the baseline UPR the MTWV values were close to 0.01 (not particularly usable results). For TLPR the dictionary-based result was about 0.15, while the example-based was much lower at 0.11. The adapted UPR, using the MFCC+fMLLR AM (as used for Figure 8 results), dictionary-based achieves 0.09, while example-based and 0.08. We also computed the MTWV score using a better acoustic model (for which we did not get the DET curves. That model was using NMC features for the AM, and after adaptation achieve 0.15 MTWV for dictionary-based search and 0.12 for example-based search.

These numbers are considerably lower than what is reported in the Babel evaluations for Amharic. We note though that the Babel results used a considerable amount of training data with orthographic representations, and the recognition systems are word-based or syllable-based (but not phonetic).

4.3 Active Learning – First Iteration of Feedback Elicitation

Active learning is a special case of semi-supervised learning in which the system is able to interactively query a user to obtain desired input at certain data points. In situations where unlabeled data is available but manual labor limited active learning can be used to query user for labels.

In our specific case, dealing with languages without known/consistent orthography, it is desirable to obtain user input without asking for orthographic transcriptions. The approach we take is to ask the user for confirmation on the output of the LUD task. The confirmations are used as means to adjust our confidence on the system output and decide which data to use for follow up system adaptation. So this is a "looser" use of the term "active learning".

In the following section we define the feedback elicitation tasks that we designed and implemented to get input from speakers without the use of orthographic labels. We then describe how we selected which data to present to users to obtain input and how we attempted to use the

output to form annotations that can help adapt and improve the system. The approach so far did not yield positive results (no further improvements after the bootstrap adaptation step) but we were not able to use enough users for enough time to get the feedback we wanted. We outline the ideas explored under this project, but also several ideas that can be employed in the future to yield improvements.

4.3.1. Feedback Elicitation Tasks

The goal of the feedback elicitation tasks is to use non-expert native speaker knowledge (no transcriptions) to inform system development. We use the LUD output and select portions that can be presented to user to provide feedback, either after the initial bootstrap step of in subsequent iteration steps. Three types of tasks were originally envisioned:

- "IsWord": judge if a hypothesized audio segment is indeed a word;
- "SameWord": judge if two different audio segments are instances of the same word;
- "Respeak": respeak word(s) presented in an audio segment.

The Respeak task was originally designed to use for low-confidence audio segments in order to get more reliable instances. However, in the course of the study, it was only used during the bootstrap step, for random phrases, to produce clean, broadband versions of audio in the training corpus, which in turn could reliably be transcribed by non-expert linguists (as described in section 4.2.2. and 4.2.3.). The tool was also used to collect additional respoken audio for about 3K Amharic and 455 Pashto utterances. We hypothesize that these cleaner examples should benefit lexical discovery and plan to run experiments to this effect in the future. The data has so far not been transcribed but is available for other researchers in the community. Neither of this efforts was part of the active learning circle that is the scope of this section.

Both the IsWord and SameWord tasks have been employed as originally intended; screenshots are presented below, in Figures 9 and 10 respectively.

Figure 9: Screen capture of IsWord task

Figure 10: Screen capture of SameWord task

4.3.2. IsWord/SameWord Utility

One of the challenges of the LUD discovery process is the great number of produced lexical hypotheses. As is shown in step 1 of Table 7 below, the process produces 1.7 million and 5.1 million tokens for Amharic and Pashto, respectively, representing over 11,000 and 27,000 different word types, respectively.

In order to select from this set a sample that is feasible to present to speakers and is likely to yield confirmed lexical units, we apply a sequence of steps as shown in Table 5. First we apply

the filter described in step 2: we select types that occur at least 100 times, that have at least five phones (i.e. about 3 syllables) and conform to the language phonotactic constraints. The resulting set of 28,000 Amharic tokens (172 types) and 106,000 Pashto tokens (671 types) is still too numerous to submit as a whole to native speakers for confirmation. Therefore, the set of lexical hypotheses was filtered further according to acoustic similarity. We used the similarity score produced from Jansen's tool (step 3) and applied a threshold (adjusted empirically to keep a feasible number of candidates). The selected set of 149 (Amharic) and 565 (Pashto) lexical unit tokens were presented to users for feedback (step 4), resulting in 54 (21 types) and 104 (52 types) confirmed lexical units, for Amharic and Pashto respectively.

The fact that we got some confirmed lexical items does not readily allow us to use this information to improve the system. Only the boundary information is really useful at this stage (which we are not currently exploiting in the model). In order to adapt the UPR we wanted some evidence that the discovered representation for the lexical unit is correct. That was the reason behind the design of the SameWord task. It was our assumption that if the same/similar pattern was confirmed multiple times to be the same lexical item, then we had evidence that the pattern was indeed the correct (or close enough) phonetic representation for the item.

Table 7: Lexical discovery confirmation steps and numbers of candidate tokens and types at each step

Steps		Amharic		Pashto	
		#Tokens	#Unique patterns	#Tokens	#Unique patterns
1	Total hypothesized	1.7M	11K	5.1M	27K
2	Filter: # occurences > 100 Duration > 5 phones Syl constraint compliant	28K	172	106K	671
3	Jansen's similarity score > threshold (threshold selected empirically)	149	34	565	139
4	Confirm via "IsWord" task	54	21	103	52
5	Form word pairs from confirmed tokens	25x2	21	29x2	21
6	Confirm via "SameWord" task	Skipped		10	8
5a	Increase step 3 threshold on step 4 confirmed types, to increase # pairs	4087x2	21	4900x2	52
6a	Confirm via "SameWord" task	301/720	11	14/280	8

In order to verify if different occurrences of similar patterns correspond to the same lexical unit, we constructed pairs (step 5) from the confirmed tokens in step 4 and submitted to users for the SameWord task (step 6). The pairs at this stage consisted of identical patterns found in different audio segments, but in follow up iterations we planned to expand this to include similar patterns as well. Given the low rate at which the original procedure yielded confirmed lexical units, we ended up with very few pairs, so we decided to employ a different strategy (which we denote as steps 5a and 6a in Table 7). We took representative samples of the 21 Amharic and 52 Pashto type patterns confirmed in step 4 and paired them with tokens with acoustic similarity score

below the threshold defined in step 3 (basically we relaxed the acoustic similarity constraint and allowed pairing with tokens in different acoustic similarity clusters). This method produced around 4,000 and 5,000 pairs for Amharic and Pashto, respectively. We did not have enough time in the project to complete the confirmation step for "SameWord" task, but the confirmations so far as shown in the last line of Table 7.

The segments confirmed as same words in steps 6 and 6a of the above described process were utilized for adaptation (see section 4.3.3.) using the hypothesized phonetic transcriptions as reference. Furthermore, the word boundary information for the confirmed lexical items was important information that follow-up iterations in the lexical discover process may utilize.

As shown in Table 8, the initial strategy of using the SameWord task to confirm lexical units yielded almost binary results: for a given clyster pattern (different audio segments with the same phonetic pattern), either almost none of the pairs were confirmed or virtually all did. This strategy was quite slow for the confirmation of different word types, since all of the tokens of a given type were examined before the next one was seen. In order to speed up the discovery of different types, we subsequently reduced the pair count to 10 per lexical type. The confirmed pairs and rates of confirmation obtained from this new process are shown in Table 9.

Table 8: Confirmation rates for sample lexical unit types, initial approach

Cluster pattern	# confirmed pairs	# pairs presented	Confirmation rate
_a_m_a_n_a_	1	204	0.005
_ax_t_a_m_a_	1	125	0.008
_n_a_m_ax_n_	127	188	0.67
_m_ax_n_a_m_ax_n_	140	150	0.93

Table 9: Confirmation rates for sample lexical unit types, modified approach

Cluster pattern	# confirmed pairs	# pairs presented	Confirmation rate
a_n_a_s_a_	3	3	1
_a_m_ax_n_d_	4	10	0.4
_a_s_a_l_a_	1	10	0.1
_a_n_d_a_n_	7	10	0.7
_n_a_m_ax_n_a_	4	10	0.4
_a_m_ax_n_ax_	5	10	0.5
_n_t_ax_n_a_	8	10	0.8

It seems from Table 9 that even with just 10 examples there is a trend of either mostly confirming or mostly rejecting the pairs. Once this trend is reliably established for a cluster (say with 20-30 examples) no more pairs for that pattern cluster need to be examined; the system will confirm or reject automatically the rest of the pairs for that cluster, and then move on to the next.

After the initial LUD step, the confirmed lexical units were resubmitted to speakers for translation. The task was completed via the "IsWord" interface where the translation could be added as a comment. This way the users could again listen to the segment both in isolation and in context, since context is useful for determining the correct translation for the word.

The results for Amharic are summarized in Table 10, to demonstrate the type of words discovered. It is worth noting that three of the patterns correspond to the same word in English, which is to be expected if the target language has morphological variation without correlates in English. It is also expected if the cluster patterns are similar enough and may be due to recognition errors of the UPR. A follow up merging step using the SameWord task can ask speakers to confirm whether similar patterns should be merged as the same unit.

It is also encouraging to see that the discovered lexical units appear to correspond to frequent words. The selection of the units presented to speakers was such that it targeted frequent words (i.e. with more than 100 occurrences in the corpus). The duration filter made sure that we avoid short, function words. Getting frequent content words is important for determining the topic of the document (of course several non-informative words are also discovered).

Table 10: Translations for confirmed lexical units

Cluster pattern	Translation	Confirmation rate
_a_m_a_n_a_	between us	0.005
_ax_t_a_m_a_	in addition	0.008
_n_a_m_ax_n_	something	0.67
_m_ax_n_a_m_ax_n_	something	0.93
a_n_a_s_a_	thirty	1
_a_m_ax_n_d_	-- not sure --	0.4
_a_s_a_l_a_	while working	0.1
_a_n_d_a_n_	some	0.7
_n_a_m_ax_n_a_	something	0.4
_a_m_ax_n_ax_	in faith	0.5
_n_t_ax_n_a_	what's his name	0.8

Similarly, Table 11 illustrates the translations of the discovered units for Pashto. The picture for Pashto was somewhat less clear. For instance, there was a higher rate of lexical units that the native speakers were not able to provide a translation for or even rejected despite having previously confirmed (we did not include those in the table). We also observe, as in Amharic, that the same translation is given to multiple different pattern clusters. How much of this is due to morphological variation, lexical synonymy or other factors related to UPR performance, remains to be explored. Still, there were very useful (and potential topic relevant) units discovered, such as marriage, Taliban, patients, Karmal (name) etc.

Table 11: Translation for confirmed lexical units in Pashto

# clusters	translation	#clusters	translation
8	Marriage/wedding	4	I am going
6	four	5	certainly
7	taliban	1	Afghani (money)
7	past	3	middle
3	patients	1	rain
>10	oneself	1	9000
4	news	1	Karmal (name of known person)

4.3.3. UPR Adaptation Based On User Feedback

The proposed idea for iterative improvement of the UPR system was to use for adaptation purposes the patterns confirmed by the native speakers as lexical units. We had two choices: adapt only on the segments of the confirmed units themselves, or adapt on the whole utterances containing the units. Since the audio data corresponding to only the segments of the confirmed units was very little (a couple of minutes), we decided to utilize the whole utterances, making the assumption that if an utterance contains confirmed units, then it has a higher average recognition accuracy, relative to the rest of the data.

The impact of the AM adaptation based on the user feedback, is illustrated in Figure 11 (Amharic only results). We started with the `fMLLR` AM (unsupervised adapted MFCC model) along with the result from the bootstrap adaptation step which used the `Hand trans.` (small amount of manual phonetic transcriptions as described in section 4.2.4.). Then we compared those results with the result after using the `Discovered` adaptation data, i.e. the utterances with confirmed discovered lexical units (with the automatically derived phonetic transcription as adaptation reference) and then the combined set of `Discovered` and `Hand trans.` together (`Disc.+Hand`). There is some promise, in the fact that adapting on the `Discovered` set gave a similar improvement as the `Hand trans`, without needing any manual transcriptions, but the combination of the two did not show any additional improvement. That shows that with the manual transcriptions we have gained significant enough improvement in performance; that the additional adaptation, obtained at the first iteration of active learning, cannot improve upon. We do know that there is more to gain with additional data (from the `Forced-30'` result in Table 5), but it apparently needs more data. It is possible that using the transcription from the recognition system as reference (we know it still has a high TPER) is limiting the effect of adaptation at this stage, and we need much more data to overcome impact of the errors in the adaptation references.

We also adapted the language model by interpolating the bootstrap-adapted LM (used in Table 5) with an LM trained using the hypotheses of the utterances containing discovered words. In the language model, the discovered words were represented as whole words to create a hybrid word/phone model. The words were converted to their pronunciation phone sequence for

obtaining the TPER results. This result is shown in the last bar of Figure 11, and it seems it is not better than just using the bootstrap-adapted LM alone.

Overall the results from the experiments summarized in Figure 11 were not positive. We could not get improvements over the bootstrap adapted system using the feedback data for adaptation purposes as described in this section. But the fact that we can get improvements over the unadapted system is a very positive indicator that we are moving the right direction. The data we are generating are useful for adaptation purposes – but we need to get more input from speakers and more confirmed units to make an impact. If we have enough confirmed data we can also try using only the confirmed segments (not the whole utterance) to limit the effect of errors in the utterance phonetic transcription.

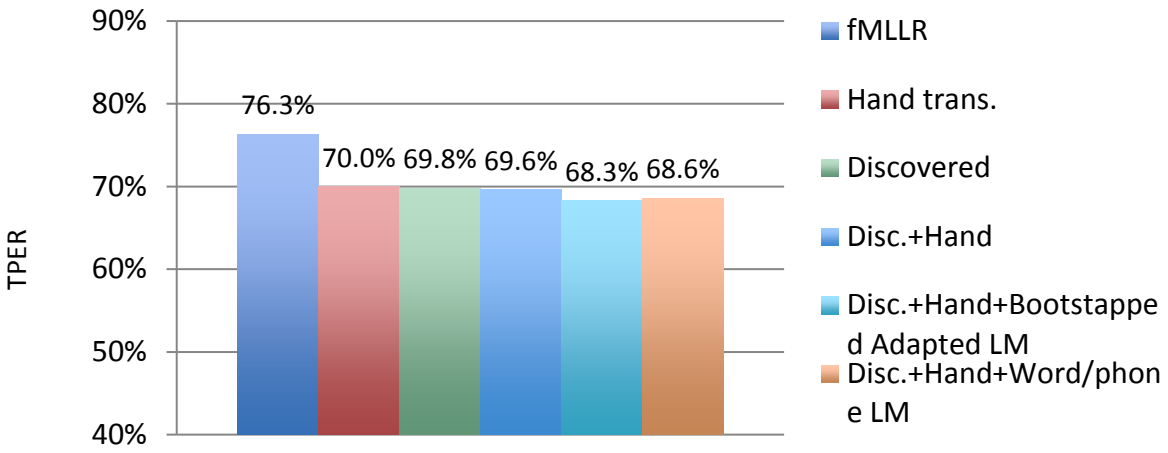

Figure 11: Effect of adaptation based on user feedback (Amharic)

4.3.4. General Lessons Learned from Speakers using the Feedback Elicitation Tool

After a number of native speakers of the target languages started using the elicitation tool, a number of important points became apparent, some of which prompted us to make modification to the tool itself. For instance, initially there was no option to "skip" a given judgment task, which often forced speakers into false choices, especially if the audio was judged not to even be in-language or unintelligible for other reasons. Similarly, we decided to include a comment function allowing speakers to leave notes about special cases or other observations about the data. This functionality also proved helpful in eliciting translations for the discovered lexical units. In other respects, the design of our elicitation tasks already anticipated important requirements of the task. For instance, the inclusion of audio playback with short (< 1 sec) windows around the segment of interest proved vital in enabling speakers to make appropriate judgments about lexical units given the surrounding linguistic context.

4.4 Next Steps for Additional Improvements

4.4.1. Additional Utility of Speaker Judgments

So far, we have concentrated on lexical unit hypotheses that native speakers have positively confirmed to be lexical units. However, it is also possible to use rejected lexical hypotheses in system development. There are, to be sure, a number of possible reasons why an audio segment presented to a native speaker is rejected as a lexical unit. This is reflected in the feedback that users can give. Thus, if users judge the audio segment to contain more than one word, we can assume that at least the lexical boundaries at the beginning and end of that segment are valid. Segments with this judgment could in turn be presented to the respeaking task to ascertain the lexical boundaries within the original segment.

When the speaker rejects the candidate lexical unit outright, a number of options can be pursued. For instance, if other hypotheses from the UPR further down the n-best list are used for the next iteration, alternative phone string output with a different similarity profile can then be submitted to speaker verification in the next iteration round. Regions of repeated rejection are prime candidates for the respeaking task in order to yield more accurate phone representations with more successful lexical confirmation.

4.4.2. Additional Utility of Respoken Data

As has been mentioned previously, we have collected, via the re-speaking task, a sizable collection of clean versions of some of the original audio data. So far, this data has not been brought to bear on the central tasks of the project, but a promising next step involves the use of this data in the lexical discovery procedure. A reasonable expectation is that audio of this kind will lend itself to more accurate pattern matching. Furthermore, the fact that one modality of the respoken utterances contains explicit lexical boundaries so far has not been sufficiently used to support lexical unit discovery. This is an especially promising area to explore since, in our experience, native speakers have found the inclusion of word boundary a relatively easy task.

The success of transcribing target language data, once clean versions with lexical breaks are elicited, has spurned discussions with other researchers who have similarly targeted the problem of transcription without native speaker experts, but have taken different approaches towards this goal. We hope to soon share data collected under the different approaches (from us and other researchers) and compare the efficacy of the different strategies.

4.4.3. Further Lexical Discovery System Improvements

In future work, we plan to explore phone confusion matrices from phone recognizer as well to enhance the fuzzy matching, in addition to the linguistically defined confusable phones. For example, we can use the Respeak activity to get more instances for same word and collect phone confusion statistics across different instances of same words.

We also plan to tightly integrate linguistic knowledge for word/syllable composition into the lexical unit discovery algorithm. Currently, we only employed them in the UPR and applied the linguistic constraints as a post-processing filtering step for candidate selection after our LUD approach.

Additionally, we would like to explore other modeling approaches, such as noisy-channel model and adaptor grammar ([3]), for modeling variability (i.e., different phone sequences for same word) and modeling the synergy between discovering sub-lexical and lexical units.

4.4.4. Further improvements for UPR Acoustic Model and Adaptation

In future work on this area, there are a number of additional improvements that could be made to improve UPR performance. First, we could improve the inventory of background training languages. Any one language on its own is likely to show a large improvement, but using a much larger pool of languages could potentially give improvements. Furthermore, we currently limit the amount of data used from resource rich languages (such as English) to avoid drowning out languages with less data. Interpolation of language specific acoustic models could be explored to remedy this problem, while making use of more data from resource rich languages.

Another path to pursue is to use a system based on automatically derived units. Derived units could be discovered using Kohonen neural networks or a Hierarchical Bayesian Model. System combination of linguistic and automatically derived acoustic models could also be tried.

Finally, more strategies for adapting the system using user feedback need to be explored. One potential strategy is to boost models based on a comparison between recognizer output on the respoken and original recordings. Another is to create target language-specific keyword spotting models directly from the discovered words.

5. CONCLUSIONS

The goal of this work has been to develop an ASLP system, such as a speech recognition or KWS system without using any orthographically transcribed training data. This has been a very challenging task, and it involves an LUD component where the system is iteratively learning the vocabulary of the language and is using lightly supervised annotations on the data to iteratively augment and confirm the discovered vocabulary.

In this report we demonstrated promising results from early stages of system testing. Specifically we have shown that the proposed approach is capable of successfully discovering useful terms. Among the main achievements of the project we consider the following three especially worthy of note:

(1) we successfully demonstrated the use of speaker/linguist input in improving the system at the bootstrap phase.
(2) we developed a utility that allows speakers to provide feedback on the units that the system has been able to discover, and
(3) we have developed a setup for the collection of cleaner versions of the data which enables non-experts of the language to contribute to the human transcription effort. We have started collaborative efforts with other researchers to share this data and investigate synergies with alternate approaches.

In addition, we implemented and delivered a first version of example-based KWS, based on universal phone recognition.

The development of the infrastructure for this work has been very challenging and, at the same time, very informative. It contributed greatly to our understanding of how to elicit information from speakers without one-on-one interactions with linguists. It further sharpened our

understanding of what types of additional tasks can be used for eliciting additional information. Finally, it helped us better understand how to interpret and how much to trust input from speakers.

One of the many challenges of the project has been to ensure availability of native speakers to provide feedback to the system when needed. For the target languages (Amharic and Pashto) it was actually pretty easy to find speakers in the area. But as the timelines for different tasks got shifted to later dates, the speakers we found became less available, and the hiring process was not fast enough to get people on board when needed. So we should have hired a larger pool of people at the beginning, to have more backup options. Letting people work remotely through a web interface allows more people to work at their own pace to contribute data to the system, but makes them less committed to their task, than working on-site. The infrastructure of the web interface has matured greatly over the course of the project and is now ready to be used for additional iterations with speakers and natural data.

The results obtained at the early stages of the system were very encouraging. We achieved, after very limited adaptation of the UPR system, LUD performance very close to that of acoustic pattern matching with the most robust features. There are several improvements that can be made to the system (as discussed in section 4.4) to further improve performance. We believe it is critical to incorporate some of the features of recent work, for example a noisy-channel model (as in [3]), for modeling variability (i.e., different phone sequences for same word) and adaptor grammar for modeling the synergy between discovering sub-lexical and lexical units.

We believe the work performed under the present project has utility for follow-on efforts. In particular, we see relevance to a number of areas of spoken language technology, including topic discovery by means of glossing the discovered units as well as bootstrap of speech translation capabilities. We further envision the utility of the present work in multiple scenarios where there is no bandwidth for expert data annotations. The general mechanism for eliciting information from native speakers pursued here can be applied to numerous kinds of data annotations and can lead to more efficient creation of databases needed for spoken language system development.

Even though it seems that with our approach we cannot improve over that acoustic-match results, as mentioned in section 4.1.1., our approach has the benefit that the discovered units can be represented in a traditional phonetic-based lexicon, can be read and interpreted by linguists and can be used for follow up ASLP applications that have been developed to rely on a lexicon representation of the lexical units.

6. REFERENCES

[1] A. Jansen, K. Church, and H. Hermansky, “Towards spoken term discovery at scale with zero resources,” in Proceedings of Interspeech, 2010.
[2] H. Kamper, A. Jansen, S. King, and S. Goldwater, “Unsupervised lexical clustering of speech segments using fixed dimensional acoustic embeddings,” in Proceedings of SLT, 2014.
[3] C. Lee, T. O’Donnell, and J. Glass, “Unsupervised lexicon discovery from acoustic input,” Transactions of the Association for Computational Linguistics, vol. 3, pp. 389–403, 2015.
[4] S. Tsakalidis, R.-C. Hsiao, D. Karakos, T. Ng, S. Ranjan, G. Saikumar, L. Zhang, L. Nguyen, R. Schwartz, and J. Makhoul, “The 2013 BBN vietnamese telephone speech keyword spotting system,” in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, 2014, pp. 7829–7833.
[5] A. Ragni, K. M. Knill, S. P. Rath, and M. J. F. Gales, “Data augmentation for low resource languages,” in INTERSPEECH, 2014.
[6] M. A. Carlin, S. Thomas, A. Jansen, and H. Hermansky, “Rapid evaluation of speech representations for spoken term discovery,” in Proceedings of Interspeech, 2011.
[7] Z. Harris, “From phoneme to morpheme,” Language, vol. 31, pp. 190–222, 1995.
[8] M. Johnson, T. L. Griffiths, and S. Goldwater, “Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models,” in Advances in Neural Information Processing Systems, 2006, pp. 641–648.
[9] S. Goldwater, T. L. Griffiths, and M. Johnson, “A Bayesian framework for word segmentation: Exploring the effects of context,” Cognition, vol. 112, pp. 21–54, 2009.
[10] B. Borshinger and M. Johnson, “Exploring the role of stress in Bayesian word segmentation using adaptor grammars,” Transaction of ACL, vol. 2, pp. 93–104, 2014.
[11] M. Hulden, "Foma: a finite-state compiler and library", Association for Computational Linguistics, pp. 29–32. 2009.
[12] J. van Hout, V. Mitra, Y. Lei, D. Vergyri, M. Graciarena, A. Mandal and H. Franco, “Recent Improvements in SRI’s Keyword Detection System for Noisy Audio,” in Proc. of Interspeech, pp. 1727-1731, Singapore, 2014.
[13] “Openkws13 keyword search evaluation plan”.

LIST OF ACRONYMS

Acronym	Definition
ALAPUL	Active Learning for Automatic Processing of Unwritten Languages
ASLP	Automatic Spoken Language Processing
AM	Acoustic Model
BGLM	BackGround Language Model
BN-DAE	BottleNeck Deep AutoEncoder
CNN	Convolutional Neural Network
DET	Detection Error Tradeoff
DNN	Deep Neural Network
DOC	Damped Oscillator Coefficients
DTW	Dynamic Time Warping
fMLLR	Feature space Maximum Likelihood Linear Regression
FST	Finite State Transducer
KWS	KeyWord Spotting
LM	Language Model
LUD	Lexical Unit Discovery
MAP	Maximum A Posteriori
MFB	Mel FilterBank
MFCC	Mel-frequency cepstral coefficients
NMC	Normalized Modulation Coefficients
PRB	Precision-recall breakeven
PRB_{SP}	Precision-recall breakeven, same speaker
PRB_{DP}	Precision-recall breakeven, different speaker
SAD	Speech Activity Detection
TPER	Time-mediated phone error rate
TLPR	Target Language Phone Recognizer
UPR	Universal phone recognizer
XFST	Xerox Finite State Tool

Appendix A: UPR phonetic inventory

UPR-based phonetic inventory for each of the training and test languages.

SRI Universal	Phonetic Definition	IPA	Training Languages								Test Languages	
			English	Spanish	Mandarin	Egyptian Arabic	Dari	Iraqi Arabic	Assamese	Bengali	Pashto	Amharic
p	voiceless bilabial stop	p pʰ pʼ	X	X	X		X		X	X	X	X
b	voiced bilabial stop	b β	X	X	X	X	X	X	X	X	X	X
t	voiceless coronal stops	t tʰ tʼ tˤ ʈ ʈʰ	X	X	X	X	X	X	X	X	X	X
d	voiced coronal stop	d dˤ ɖ	X	X	X	X	X	X	X	X	X	X
k	voiceless dorsal stops	k kʰ kʼ q	X	X	X	X	X	X	X	X	X	X
g	voiced velar stop	g	X	X	X	X	X	X	X	X	X	X
Q	glottal stop	ʔ				X	X	X				
f	voiceless labiodental fricative	f	X	X	X	X	X	X	X	X	X	X
B	voiced bilabial fricative	β		X								
v	voiced labiodental fricative	v	X		X				X	X		X
T	voiceless interdental fricative	θ	X					X				
D	voiced interdental fricative	ð ðˤ	X	X		X		X				
s	voiceless alveolar fricative	s sʰ sˤ sʼ	X	X	X	X	X	X	X	X	X	X
z	voiced alveolar fricative	z	X	X		X	X	X	X	X	X	X
S	voiceless retroflex/post-alveolar fricative	ʂ ʃ ʃʰ	X	X	X	X	X	X	X	X	X	X
Z	voiced retroflex/post-alveolar fricative	ʐ ʒ	X				X		X	X	X	X
x	voiceless velar fricative	x		X	X	X	X	X	X		X	
G	voiced velar fricative	ɣ		X		X	X	X			X	
h	glottal fricatives, voiceless pharyngeal fricative	h ɦ ʜ	X			X	X	X	X	X	X	X
Hv	voiced pharyngeal fricative	ʕ				X		X				
44	alveolar trill	r		X		X						
4	alveolar flap	ɾ	X	X			X	X	X	X	X	X
m	bilabial nasal	m	X	X	X	X	X	X	X	X	X	X
n	coronal nasal	n ɳ	X	X	X	X	X	X	X	X	X	X
J	palatal nasal	ɲ		X								X
N	velar/uvular nasal	ŋ ɴ	X		X		X		X	X	X	
l	coronal lateral approximant	l ɭ	X	X	X	X	X	X	X	X	X	X
w	labial approximants	W ʊ ɥ	X	X	X	X	X	X	X	X	X	X

SRI Universal	Phonetic Definition	IPA	Training Languages								Test Languages	
			English	Spanish	Mandarin	Egyptian Arabic	Dari	Iraqi Arabic	Assamese	Bengali	Pashto	Amharic
r	coronal approximant	ɹ ɻ	X		X				X	X	X	
j	palatal approximant	j	X	X	X	X	X	X	X	X	X	X
ts	voiceless alveolar affricate	ts tsʰ			X						X	
dz	voiced alveolar affricate	dz			X						X	
tS	voiceless retroflex/post-alveolar affricate	ʈʂ ʈʂʰ tʃ tʃʰ tʃ͡	X	X	X		X	X	X	X	X	X
dZ	voiced retroflex/post-alveolar affricate	ɖʐ dʒ	X	X	X	X	X	X	X	X	X	X
i	close front unrounded vowel	iː i i̯	X	X	X	X	X	X	X	I	X	X
I	near-close front unrounded vowel	ɪ	X			X	X	X			X	
e	long close-mid front unrounded vowel	eː e eʲ	X	X		X	X	X	X	X	X	X
E	open-mid front unrounded vowel	ɛ	X		X		X		X			
ae	near-open front unrounded vowel	æː æ	X			X	X			X		
u	close back rounded vowel	uː u	X	X	X	X	X	X	X	X	X	X
U	near-close back rounded vowel	ʊ	X					X			X	
o	close-mid back rounded vowel	oː o oʷ	X	X	X	X	X	X	X	X	X	X
O	open-mid back rounded vowel	ɔ	X		X		X		X	X		
ax	mid unrounded vowel	ɨ ə	X		X					X	X	X
er	rhoticized mid central unrounded vowel	ɚ	X		X							
y	close front rounded vowel	y			X							
V	open-mid back unrounded vowel	ʌ	X									
a	open unrounded	ɑː ɑ a	X	X	X	X	X	X	X	X	X	X
aj		aʲ ɑʲ əʲ	X	X	X	X	X		X	X	X	X
Oj		ɔʲ oʲ uʲ	X	X			X		X	X	X	
ew		eʷ		X								
aw		aʷ ɑʷ	X	X	X	X	X			X	X	X

Appendix B: Phone map from Amharic to Universal Phone Set

Amharic Phone	Universal Phone(s)	Phonetic description
Stops		
p	p	voiceless bilabial stop
p>		voiceless bilabial ejective/glottalized stop
b	b	voiced bilabial stop
t	t	voiceless alveolar stop
t>		voiceless alveolar ejective/glottalized stop
d	d	voiced alveolar stop
k	k	voiceless velar stop
k>		voiceless velar ejective/glottalized stop
g	g	voiced velar stop
?	*removed*	glottal stop
Fricatives		
f	f	voiceless labiodental fricative
v	v	voiced labiodental fricative
s	s	voiceless alveolar fricative
s>		voiceless alveolar ejective/glottalized fricative
z	z	voiced alveolar fricative
S	S	voiceless palato-alveolar fricative
Z	Z	voiced palato-alveolar fricative
h	h	voiceless glottal fricative
Affricates		
tS	tS	voiceless palato-alveolar affricate
tS>		voiceless palato-alveolar ejective/glottalized affricate
dZ	dZ	voiced palato-alveolar affricate
Nasals		
m	m	bilabial nasal
n	n	alveolar nasal
J	J	palatal nasal
Approximants and glides		
l	l	alveolar lateral approximant
r	r	alveolar approximant
j	j	palatal glide
w	w	labiovelar glide
Vowels		
i	i	high front vowel
e	e	mid front vowel
u	u	high back vowel
o	o	mid back vowel
a	a	low vowel
1	ax	high central vowel
@		mid central vowel
Labialized stops		
pw	p w	voiceless labialized bilabial stop

bw	b w	voiced labialized bilabial stop
tw	t w	voiceless labialized alveolar stop
t>w		voiceless labialized alveolar ejective/glottalized stop
dw	d w	voiced labialized alveolar stop
kw	k w	voiceless labialized velar stop
k>w		voiceless labialized velar ejective/glottalized stop
gw	g w	voiced labialized velar stop
Labialized fricatives		
fw	f w	voiceless labialized labiodental fricative
vw	v w	voiced labialized labiodental fricative
sw	s w	voiceless labialized alveolar ejective/glottalized fricative
s>w		voiceless labialized alveolar fricative
zw	z w	voiced labialized alveolar fricative
Sw	S w	voiceless labialized palato-alveolar fricative
Zw	Z w	voiced labialized palato-alveolar fricative
hw	h w	voiceless labialized glottal fricative
Labialized affricates		
tSw	tS w	voiceless labialized palato-alveolar ejective/glottalized affricate
tS>w		voiceless labialized palato-alveolar affricate
dZw	dZ w	voiced labialized palato-alveolar affricate
Labialized nasals		
mw	m w	labialized bilabial nasal
nw	n w	labialized alveolar nasal
Jw	J w	labialized palatal nasal
Labialized Approximants		
lw	l w	labialized alveolar lateral approximant
rw	r w	labialized alveolar approximant

Appendix C: Phone Map from Pashto to Universal Phone Set

Pashto Phone	Universal Phone	Phonetic Description
Stops		
p	p	voiceless bilabial stop
b	b	voiced bilabial stop
t	t	voiceless dental stop
t`		voiceless retroflex stop
d	d	voided dental stop
d`		voiced retroflex stop
k	k	voiceless velar stop
q		voiceless uvular stop
g	g	voiced velar stop
?	Q	glottal stop
Fricatives		
f	f	voiceless labiodental fricative
s	s	voiceless alveolar fricative
z	z	voiced alveolar fricative
s`	S	voiceless retroflex fricative
S		voiceless palato-alveolar fricative
C		voiceless palatal fricative
z`	Z	voiced retroflex fricative
Z		voiced palato-alveolar fricative
j\		voiced palatal fricative
x	x	voiceless velar fricative
G	G	voiced velar fricative
h	h	voiceless glottal fricative
Affricates		
ts	ts	voiceless alveolar affricate
dz	dz	voiced alveolar affricate
tS	tS	voiceless palato-alveolar affricate
dZ	dZ	voiced palato-alveolar affricate
Nasals		
m	m	bilabial nasal
n	n	dental nasal
n`		retroflex nasal
N	N	velar nasal
Approximants, taps, and glides		
l	l	alveolar lateral approximant
r`	r	retroflex approximant
4	4	dental flap
j	j	palatal glide
w	w	labiovelar glide
Monophthong vowels		
@	ax	mid central vowel
A	a	low back vowel
a		low front vowel
e	e	mid front vowel
o	o	mid back vowel

i:	i	tense high front vowel
i	I	lax high front vowel
u:	u	tense high back vowel
u	U	lax high back vowel
Diphthong vowels		
A j	aj	low back to high front diphthong
a j		low front to high front diphthong
@ j		mid central to high front diphthong
A w	aw	low back to high back diphthong
a w		low front to high back diphthong
@ w		mid central to high back diphthong

Appendix D: Foma rules for Amharic

```
################################################
## ## Vowels

## Basic Vowels
define V [{_a_}|{_e_}|{_i_}|{_o_}|{_u_}|{_ax_}|{_aw_}|{_aj_}];

## Complex Vowels
define ComplexV [{_a_}{_a_}|
{_a_}{_aj_}|
{_a_}{_aw_}|
{_a_}{_e_}|
{_a_}{_o_}|
{_ax_}{_a_}|
{_ax_}{_ax_}|
{_ax_}{_aj_}|
{_ax_}{_aw_}|
{_ax_}{_e_}|
{_ax_}{_i_}|
{_ax_}{_o_}|
{_ax_}{_u_}|
{_aj_}{_e_}|
{_i_}{_aj_}|
{_i_}{_e_}|
{_i_}{_a_}|
{_i_}{_e_}|
{_u_}{_a_}|
{_u_}{_aw_}|
{_e_}{_e_}|
{_e_}{_a_}|
{_e_}{_i_}|
{_e_}{_w_}|
{_i_}{_aj_}|
{_i_}{_i_}|
{_i_}{_o_}|
{_i_}{_w_}|
{_o_}{_o_}|
{_o_}{_a_}|
{_u_}{_a_}|
{_u_}{_e_}|
{_u_}{_aw_}|
{_u_}{_u_}];

## Vowels can be basic or complex
define VB V | ComplexV;
################################################
## ## Consonants

## Basic Consonants
define CB [{_t_}|{_l_}|{_m_}|{_n_}|{_j_}
       |{_k_}|{_s_}|{_4_}|{_b_}|{_w_}
       |{_d_}|{_g_}|{_tS_}|{_h_}|{_S_}
       |{_J_}|{_f_}|{_z_}|{_dZ_}
       |{_p_}|{_v_}|{_Z_}];

## Consonants in Syllable Onsets
define SimpleOnset [ CB - { 44 } - { w }];
define ComplexOnset
[{_d_}{_w_}|{_t_}{_w_}|{_s_}{_w_}|{_m_}{_w_}|{_f_}{_w_}|{_4_}{_w_}|{_g_}{_w_}|{_k_}{_w_}|{_s_}
{_t_}|{_s_}{_k_}];
```

```
## Consonants in Syllable Codas
define SimpleCoda [ CB - { 44 } ];
define ComplexCoda [{_4_}{_S_}|
{_4_}{_b_}|
{_4_}{_d_}|
{_4_}{_f_}|
{_4_}{_g_}|
{_4_}{_h_}|
{_4_}{_k_}|
{_4_}{_l_}|
{_4_}{_m_}|
{_4_}{_n_}|
{_4_}{_p_}|
{_4_}{_s_}|
{_4_}{_t_}|
{_4_}{_v_}|
{_4_}{_z_}|
{_b_}{_4_}|
{_b_}{_S_}|
{_b_}{_d_}|
{_b_}{_h_}|
{_b_}{_k_}|
{_b_}{_n_}|
{_b_}{_s_}|
{_b_}{_t_}|
{_d_}{_S_}|
{_d_}{_k_}|
{_d_}{_n_}|
{_f_}{_k_}|
{_f_}{_l_}|
{_f_}{_t_}|
{_g_}{_4_}|
{_g_}{_d_}|
{_g_}{_n_}|
{_h_}{_h_}|
{_h_}{_k_}|
{_j_}{_d_}|
{_j_}{_s_}|
{_k_}{_l_}|
{_k_}{_m_}|
{_k_}{_s_}|
{_l_}{_S_}|
{_l_}{_b_}|
{_l_}{_d_}|
{_l_}{_f_}|
{_l_}{_g_}|
{_l_}{_h_}|
{_l_}{_k_}|
{_l_}{_m_}|
{_l_}{_n_}|
{_l_}{_s_}|
{_l_}{_t_}|
{_l_}{_v_}|
{_m_}{_4_}|
{_m_}{_S_}|
{_m_}{_b_}|
{_m_}{_d_}|
{_m_}{_n_}|
{_m_}{_p_}|
{_m_}{_s_}|
```

```
{_m_}{_t_}|
{_n_}{_S_}|
{_n_}{_b_}|
{_n_}{_d_}|
{_n_}{_f_}|
{_n_}{_g_}|
{_n_}{_k_}|
{_n_}{_s_}|
{_n_}{_t_}|
{_n_}{_z_}|
{_p_}{_t_}|
{_s_}{_S_}|
{_s_}{_d_}|
{_s_}{_k_}|
{_s_}{_l_}|
{_s_}{_t_}|
{_t_}{_b_}|
{_w_}{_s_}|
{_w_}{_t_}|
{_4_}{_dZ_}|
{_4_}{_tS_}|
{_l_}{_tS_}|
{_m_}{_tS_}|
{_n_}{_dZ_}|
{_n_}{_tS_}];

################################################################
## Simple and Complex Onsets

define Onset SimpleOnset | ComplexOnset;

## Simple and Complex Codas
define Coda SimpleCoda | ComplexCoda;

################################################################
## Syllables

## Onset-less syllables
define NOSyll VB Coda | VB;

## Open and Closed Syllables
define OpenSyll Onset VB;
define ClosedSyll Onset VB Coda;

define Syl OpenSyll | ClosedSyll;

################################################################
## Words as sequences of syllables (and optional initial onset-less syllable)
## SilWord as Word optionally preceded by silence (or laughter)

define Word NOSyll | (NOSyll) Syl+;
define SilWord ({_SIL_}|{_LAU_})* Word;

################################################################
## Syllable String

define SyllableString  ({_SIL_}|{_LAU_})*|({_SIL_}|{_LAU_})* Word ({_SIL_}|{_LAU_})* |
({_SIL_}|{_LAU_})* Word SilWord+ ({_SIL_}|{_LAU_})* ;

regex SyllableString;
```

```
random-words
quit
```

Appendix E: Foma rules for Pashto

```
## ## Vowels
## Basic Vowels
define V [{_I_}|{_Oj_}|{_U_}|{_a_}|{_aj_}|{_aw_}|{_ax_}|{_e_}|{_i_}|{_o_}|{_u_}];

## Complex Vowels
define ComplexV [{_U_}{_a_}
|{_U_}{_aj_}
|{_U_}{_i_}
|{_U_}{_u_}
|{_a_}{_U_}
|{_a_}{_a_}
|{_a_}{_e_}
|{_a_}{_i_}
|{_a_}{_o_}
|{_a_}{_u_}
|{_aj_}{_a_}
|{_aj_}{_e_}
|{_aj_}{_i_}
|{_ax_}{_a_}
|{_ax_}{_i_}
|{_ax_}{_o_}
|{_e_}{_a_}
|{_e_}{_aw_}
|{_e_}{_e_}
|{_e_}{_i_}
|{_e_}{_o_}
|{_e_}{_u_}
|{_i_}{_U_}
|{_i_}{_a_}
|{_i_}{_aj_}
|{_i_}{_aw_}
|{_i_}{_e_}
|{_i_}{_i_}
|{_i_}{_o_}
|{_o_}{_a_}
|{_o_}{_aw_}
|{_o_}{_e_}
|{_u_}{_a_}
|{_u_}{_e_}];

## Vowels can be basic or complex
define VB V | ComplexV;

################################################
## ## Consonants

## Basic Consonants
define CB [{_4_}|{_G_}|{_N_}|{_Q_}|{_S_}
       |{_Z_}|{_b_}|{_d_}|{_dZ_}|{_dz_}
       |{_f_}|{_g_}|{_h_}|{_j_}|{_k_}|{_l_}
       |{_m_}|{_n_}|{_p_}|{_r_}|{_s_}
       |{_t_}|{_tS_}|{_ts_}|{_w_}|{_x_}|{_z_}];

## Consonants in Syllable Onsets
define SimpleOnset CB;

define ComplexOnset [{_4_}{_j_}
|{_G_}{_4_}
|{_G_}{_b_}
```

```
|{_G_}{_j_}
|{_G_}{_l_}
|{_G_}{_r_}
|{_G_}{_w_}
|{_N_}{_g_}
|{_S_}{_4_}
|{_S_}{_j_}
|{_S_}{_k_}
|{_S_}{_l_}
|{_S_}{_m_}
|{_S_}{_n_}
|{_S_}{_p_}
|{_S_}{_r_}
|{_S_}{_t_}
|{_S_}{_w_}
|{_S_}{_x_}
|{_Z_}{_4_}
|{_Z_}{_G_}
|{_Z_}{_b_}
|{_Z_}{_d_}
|{_Z_}{_m_}
|{_Z_}{_w_}
|{_b_}{_4_}
|{_b_}{_h_}
|{_b_}{_j_}
|{_b_}{_l_}
|{_b_}{_r_}
|{_b_}{_w_}
|{_dZ_}{_4_}
|{_dZ_}{_G_}
|{_dZ_}{_b_}
|{_dZ_}{_m_}
|{_dZ_}{_w_}
|{_d_}{_4_}
|{_d_}{_j_}
|{_d_}{_w_}
|{_dz_}{_G_}
|{_dz_}{_m_}
|{_dz_}{_w_}
|{_f_}{_4_}
|{_f_}{_l_}
|{_g_}{_4_}
|{_g_}{_d_}
|{_g_}{_j_}
|{_g_}{_l_}
|{_g_}{_m_}
|{_g_}{_r_}
|{_g_}{_w_}
|{_k_}{_4_}
|{_k_}{_S_}
|{_k_}{_h_}
|{_k_}{_j_}
|{_k_}{_l_}
|{_k_}{_r_}
|{_k_}{_w_}
|{_k_}{_x_}
|{_l_}{_m_}
|{_l_}{_w_}
|{_m_}{_4_}
|{_m_}{_j_}
|{_m_}{_l_}
```

```
|{_m_}{_r_}
|{_n_}{_G_}
|{_n_}{_S_}
|{_n_}{_Z_}
|{_n_}{_dZ_}
|{_n_}{_d_}
|{_n_}{_j_}
|{_n_}{_m_}
|{_n_}{_x_}
|{_p_}{_4_}
|{_p_}{_S_}
|{_p_}{_j_}
|{_p_}{_l_}
|{_p_}{_r_}
|{_p_}{_w_}
|{_p_}{_x_}
|{_s_}{_4_}
|{_s_}{_j_}
|{_s_}{_k_}
|{_s_}{_l_}
|{_s_}{_m_}
|{_s_}{_n_}
|{_s_}{_p_}
|{_s_}{_t_}
|{_s_}{_w_}
|{_s_}{_x_}
|{_tS_}{_j_}
|{_t_}{_4_}
|{_t_}{_h_}
|{_t_}{_j_}
|{_t_}{_l_}
|{_t_}{_r_}
|{_t_}{_w_}
|{_ts_}{_4_}
|{_ts_}{_w_}
|{_w_}{_4_}
|{_w_}{_j_}
|{_w_}{_l_}
|{_w_}{_m_}
|{_w_}{_r_}
|{_x_}{_4_}
|{_x_}{_j_}
|{_x_}{_k_}
|{_x_}{_l_}
|{_x_}{_p_}
|{_x_}{_r_}
|{_x_}{_w_}
|{_z_}{_4_}
|{_z_}{_G_}
|{_z_}{_b_}
|{_z_}{_d_}
|{_z_}{_j_}
|{_z_}{_m_}
|{_z_}{_r_}
|{_z_}{_w_}
|{_S_}{_w_}{_j_}
|{_n_}{_d_}{_4_}
|{_p_}{_4_}{_j_}
|{_s_}{_k_}{_4_}
|{_s_}{_k_}{_w_}
|{_s_}{_p_}{_j_}
```

```
|{_s_}{_t_}{_4_}
|{_t_}{_4_}{_j_}
|{_w_}{_4_}{_j_}
|{_x_}{_w_}{_4_}
|{_x_}{_w_}{_j_}
|{_x_}{_w_}{_4_}{_j_}
|{_N_}{_g_}
|{_S_}{_t_}
];

## Consonants in Syllable Codas
define SimpleCoda CB;
define ComplexCoda [{_N_}{_g_}
|{_S_}{_t_}
|{_d_}{_z_}
|{_g_}{_s_}
|{_g_}{_z_}
|{_k_}{_s_}
|{_l_}{_d_}
|{_l_}{_z_}
|{_s_}{_t_}
|{_t_}{_s_}
|{_t_}{_z_}
|{_4_}{_k_}{_s_}
|{_4_}{_l_}{_d_}
|{_4_}{_l_}{_z_}
|{_4_}{_s_}{_t_}
|{_4_}{_t_}{_s_}
|{_4_}{_t_}{_z_}
|{_N_}{_g_}{_s_}
|{_N_}{_g_}{_z_}
|{_k_}{_s_}{_t_}
|{_k_}{_t_}{_s_}
|{_n_}{_d_}{_z_}
|{_n_}{_t_}{_s_}
|{_4_}{_S_}
|{_4_}{_b_}
|{_4_}{_dZ_}
|{_4_}{_d_}
|{_4_}{_f_}
|{_4_}{_g_}
|{_4_}{_k_}
|{_4_}{_m_}
|{_4_}{_n_}
|{_4_}{_p_}
|{_4_}{_s_}
|{_4_}{_tS_}
|{_4_}{_t_}
|{_4_}{_ts_}
|{_4_}{_x_}
|{_4_}{_z_}
|{_G_}{_z_}
|{_N_}{_g_}
|{_N_}{_k_}
|{_S_}{_d_}
|{_S_}{_k_}
|{_S_}{_t_}
|{_Z_}{_d_}
|{_Z_}{_m_}
|{_b_}{_S_}
|{_b_}{_h_}
```

```
|{_b_}{_k_}
|{_b_}{_n_}
|{_b_}{_t_}
|{_b_}{_z_}
|{_dZ_}{_z_}
|{_d_}{_z_}
|{_f_}{_4_}
|{_f_}{_l_}
|{_f_}{_t_}
|{_f_}{_z_}
|{_g_}{_d_}
|{_g_}{_m_}
|{_g_}{_s_}
|{_g_}{_t_}
|{_g_}{_z_}
|{_h_}{_4_}
|{_h_}{_d_}
|{_h_}{_f_}
|{_h_}{_l_}
|{_h_}{_n_}
|{_h_}{_s_}
|{_j_}{_4_}
|{_k_}{_4_}
|{_k_}{_S_}
|{_k_}{_d_}
|{_k_}{_s_}
|{_k_}{_t_}
|{_l_}{_b_}
|{_l_}{_d_}
|{_l_}{_f_}
|{_l_}{_k_}
|{_l_}{_m_}
|{_l_}{_p_}
|{_l_}{_s_}
|{_l_}{_t_}
|{_l_}{_x_}
|{_l_}{_z_}
|{_m_}{_d_}
|{_m_}{_p_}
|{_m_}{_s_}
|{_m_}{_z_}
|{_n_}{_Z_}
|{_n_}{_dZ_}
|{_n_}{_d_}
|{_n_}{_dz_}
|{_n_}{_f_}
|{_n_}{_k_}
|{_n_}{_r_}
|{_n_}{_s_}
|{_n_}{_tS_}
|{_n_}{_t_}
|{_n_}{_z_}
|{_p_}{_s_}
|{_p_}{_t_}
|{_r_}{_d_}
|{_r_}{_p_}
|{_r_}{_x_}
|{_s_}{_4_}
|{_s_}{_b_}
|{_s_}{_d_}
|{_s_}{_f_}
```

```
|{_s_}{_k_}
|{_s_}{_l_}
|{_s_}{_m_}
|{_s_}{_p_}
|{_s_}{_t_}
|{_t_}{_4_}
|{_t_}{_f_}
|{_t_}{_s_}
|{_t_}{_z_}
|{_w_}{_n_}
|{_w_}{_t_}
|{_x_}{_S_}
|{_x_}{_s_}
|{_x_}{_t_}
|{_z_}{_4_}
|{_z_}{_b_}
|{_z_}{_d_}
|{_z_}{_k_}
|{_z_}{_m_}];


################################################################
## Simple and Complex Onsets
define Onset SimpleOnset | ComplexOnset;
define Coda SimpleCoda | ComplexCoda;

################################################################
## Syllables
## Onset-less syllables
define NOSyll VB Coda | VB;

## Open and Closed Syllables
define OpenSyll Onset VB;
define ClosedSyll Onset VB Coda;

define Syl OpenSyll | ClosedSyll;

################################################################
## Words as sequences of syllables (and optional initial onset-less syllable)
## SilWord as Word optionally preceded by silence (or laughter)

define Word NOSyll | (NOSyll) Syl+;

define SilWord ({_SIL_}|{_LAU_})* Word;

################################################################
## Syllable String

define SyllableString  ({_SIL_}|{_LAU_})*|({_SIL_}|{_LAU_})* Word ({_SIL_}|{_LAU_})* |
({_SIL_}|{_LAU_})* Word SilWord+ ({_SIL_}|{_LAU_})* ;

regex SyllableString;

random-words
#write att PashtoSyllableStringBroad.att
quit
```